Debian Referenz

Osamu Aoki <osamu@debian.org>
Übersetzer: Jens Seidel <tux-master@web.de>
'Autoren' auf Seite 207

CVS, Mit 29. Okt 2003, 02:47:06 UTC

Zusammenfassung

Diese Debian Referenz (http://qref.sourceforge.net/) wurde geschrieben, um einen groben Überblick über das Debian System in Form einer **Referenz für die Zeit nach der Installation** zu gewinnen. Sie behandelt viele Aspekte der Systemadministration mittels **Shell-Kommando** Beispielen. Grundlegende Tutorials, Tipps und andere Informationen werden dargestellt für Themen inklusive der fundamentalen Konzepte des Debian Systems, System Installations Hinweise, Debian Paketverwaltung, den Linux Kernel unter Debian, System Tuning, Aufbau eines Gateway, Texteditoren, CVS, Programmierung und GnuPG für **Nicht-Entwickler**.

Copyright Hinweis

Copyright © 2001–2003 by Osamu Aoki <osamu@debian.org>.
Copyright (Kapitel 2) © 1996–2001 by Software in the Public Interest.

Dieses Dokument kann unter den Bedingungen der GNU General Public License Version 2 oder höher genutzt werden. (http://www.gnu.org/copyleft/gpl.html)

Es ist erlaubt, unveränderte Kopien dieses Dokuments zu erstellen und zu vertreiben, vorausgesetzt der Copyright Hinweis und diese Genehmigung bleiben auf allen Kopien erhalten.

Es ist erlaubt, veränderte Kopien dieses Dokuments unter den Voraussetzungen für unverändertes Kopieren, zu erstellen und zu vertreiben, sofern die gesamte resultierende Arbeit unter den Bedingungen einer Genehmigung identisch zu dieser, vertrieben wird.

Es ist erlaubt, Übersetzungen dieses Dokuments in eine andere Sprache, unter den obigen Bedingungen für veränderte Versionen zu kopieren und zu verteilen, mit der Ausnahme, dass diese Genehmigung übersetzt, statt im ursprünglichem Englisch, eingebunden werden kann, sofern diese Übersetzung (des Copyrights) von der Free Software Foundation genehmigt ist.

Inhaltsverzeichnis

1 Vorwort **1**

1.1 Das offizielle Dokument . . . 1
1.2 Konventionen in diesem Dokument . . . 2
1.3 Beispielskripte . . . 2
1.4 Grundlegende Einstellungen . . . 2
1.5 Grundlagen der Debian Distributionen . . . 3

2 Debian Grundlagen **5**

2.1 Die Debian Archive . . . 5
2.1.1 Verzeichnisstrukturen . . . 5
2.1.2 Debian Distributionen . . . 6
2.1.3 Die `stable` Distribution . . . 6
2.1.4 Die `testing` Distribution . . . 7
2.1.5 Die `unstable` Distribution . . . 7
2.1.6 Die `frozen` Distribution . . . 8
2.1.7 Debian Distributions-Kodenamen . . . 8
2.1.8 In der Vergangenheit verwendete Kodenamen . . . 8
2.1.9 Die Herkunft der Kodenamen . . . 9
2.1.10 Das `pool` Verzeichnis . . . 9
2.1.11 Historische Bemerkungen über Sid . . . 9
2.1.12 Heraufgeladene Pakete in `incoming/` . . . 10
2.1.13 Wiederauffinden eines älteren Paketes . . . 10
2.1.14 Architektur abhängige Verzeichnisse . . . 10
2.1.15 Der Quellcode . . . 11

2.2 Das Debian Paketverwaltungssystem 11
2.2.1 Überblick über Debian Pakete 11
2.2.2 Debian Paketformat 12
2.2.3 Namenskonventionen für Debians Paketdateinamen 13
2.2.4 Bewahren der lokalen Konfiguration 14
2.2.5 Debian Wartungsskripte 14
2.2.6 Paket Prioritäten 15
2.2.7 Virtuelle Pakete 16
2.2.8 Paketabhängigkeiten 16
2.2.9 Die Bedeutung von “pre-depends” 17
2.2.10 Paket Status 17
2.2.11 Zurückhalten von Paketen von einem Upgrade 18
2.2.12 Quellpakete 19
2.2.13 Erzeugen von Binär- aus Quellcodepaketen 19
2.2.14 Erzeugen neuer Debian Pakete 19
2.3 Aktualisierung eines Debian Systems 20
2.3.1 Methoden zum Aktualisieren eines Debian Systems 20
2.3.2 Überblick über Paketverwaltungstools 20
2.3.3 `dpkg` 20
2.3.4 APT 21
2.3.5 `dselect` 21
2.3.6 Aktualisieren eines laufenden Systems 21
2.3.7 Heruntergeladene und zwischengespeicherte `.deb` Archiv Dateien 22
2.3.8 Aufbewahren des Datensatzes für Upgrades 22
2.4 Der Debian Bootvorgang 22
2.4.1 Das `init` Programm 22
2.4.2 Runlevel 23
2.4.3 Anpassen des Bootvorgangs 23
2.5 Unterstützung von Vielfalten 24
2.6 Internationalisierung 24
2.7 Debian und der Kernel 25

2.7.1 Kompilierung eines Kernel aus Debian fremden Quellen 25
2.7.2 Tools zum Erzeugen angepasster Kernel 25
2.7.3 Alternative Boot Loader 26
2.7.4 Erzeugen von Boot Disketten 26
2.7.5 Spezielle Regeln für den Umgang mit Modulen 26
2.7.6 Deinstallation eines alten Kernel Pakets 27

3 Debian System-Installationshinweise **29**
3.1 Allgemeine Linux System-Installationshinweise 29
3.1.1 Grundlegendes zur Hardware Kompatibilität 29
3.1.2 Bestimmung der PC Hardware und des Chipsatzes 30
3.1.3 Bestimmung der PC Hardware mit Debian 30
3.1.4 Bestimmen der PC Hardware mit anderen Betriebssystemen 31
3.1.5 Ein Lilo Mythos 31
3.1.6 GRUB 31
3.1.7 Wahl der Boot Disketten 32
3.1.8 Installation 32
3.1.9 Hosts und IP im LAN 33
3.1.10 Benutzerkonten 34
3.1.11 Anlegen von Dateisystemen 34
3.1.12 DRAM Speicher Richtlinien 37
3.1.13 Swap Speicher 37
3.2 Bash Konfiguration 38
3.3 Maus Konfiguration 38
3.3.1 PS/2 Mäuse 38
3.3.2 USB Mäuse 40
3.3.3 Touchpad 41
3.4 NFS Konfiguration 42
3.5 Samba Konfiguration 42
3.6 Drucker Konfiguration 43
3.6.1 `lpr/lpd` 43

3.6.2 CUPS™ . . . 44
3.7 Andere Installations Hinweise . . . 44
3.7.1 Installation einiger weiterer Pakete nach der Basisinstallation . . . 44
3.7.2 Module . . . 45
3.7.3 Grundlegende CD-RW Einrichtung . . . 46
3.7.4 Großer Speicher und automatisches Abschalten . . . 46
3.7.5 Eigenartige Probleme beim Zugriff auf einige Webseiten . . . 47
3.7.6 Dialup PPP Konfiguration . . . 47
3.7.7 Andere Konfigurationsdateien in `/etc/` . . . 48

4 Debian Tutorials **49**
4.1 Informationsquellen . . . 49
4.2 Die Linux Konsole . . . 49
4.2.1 Login . . . 49
4.2.2 Hinzufügen eines neuen Nutzers . . . 50
4.2.3 Wie man den Rechner herunterfährt . . . 50
4.2.4 Kommandozeilen basiertes Editieren . . . 51
4.2.5 Kommandoausführung . . . 51
4.2.6 Die grundlegensten Kommandos zur Erinnerung . . . 52
4.2.7 Das X Window System . . . 52
4.2.8 Wichtige Tastaturkommandos . . . 52
4.3 Midnight Commander (MC) . . . 53
4.3.1 Installation von MC . . . 53
4.3.2 Starten von MC . . . 53
4.3.3 Dateimanager . . . 53
4.3.4 Kommandozeilen Tricks . . . 54
4.3.5 Editor . . . 54
4.3.6 Dateibetrachter . . . 55
4.3.7 Auto-Start Eigenschaften . . . 55
4.3.8 Virtuelles FTP Dateisystem . . . 55
4.4 Grundlagen des GNU/Linux Dateisystems . . . 55

4.4.1 Zugriffsrechte von Dateien und Verzeichnissen . . . 56
4.4.2 Zeitmarken . . . 57
4.4.3 Links . . . 58
4.5 Weiterführende Informationen . . . 59

5 Aktualisieren einer Distribution auf `testing` 61
5.1 Übergang von APT auf die Woody Version . . . 61
5.2 Vorbereitung des Übergangs (`stable` nach `testing`) . . . 62
5.3 Aktualisierung des Debian Systems . . . 63
5.3.1 Beste Upgrade Vorgehensweise mittels `dselect` . . . 63
5.3.2 Unerwünschte Vorgehensweise mittels `apt-get` . . . 64

6 Debian Paketverwaltung 65
6.1 Einführung . . . 65
6.1.1 Grundlegende Werkzeuge . . . 65
6.1.2 Nützliche Tools . . . 66
6.2 Grundlagen der Debian-Paketverwaltung . . . 66
6.2.1 *Aufgaben-Profile* mit `tasksel` oder `aptitude` installieren . . . 66
6.2.2 Einrichten des APT Systems . . . 67
6.2.3 `dselect` . . . 67
6.2.4 `aptitude` . . . 68
6.2.5 Die `apt-cache` und `apt-get` Kommandos . . . 68
6.2.6 Verwenden einer speziellen Version der Debian Distribution . . . 70
6.2.7 Downgrad aller Pakete zu `stable` . . . 70
6.2.8 Überblick über `/etc/apt/preferences` . . . 71
6.3 Debian Überlebensbefehle . . . 71
6.3.1 Überprüfen von Bugs in Debian und Suche nach Hilfe . . . 72
6.3.2 APT Upgrade Fehlersuche . . . 72
6.3.3 Wiederherstellung mit `dpkg` . . . 73
6.3.4 Wiederherstellung von Paketauswahldaten . . . 74
6.3.5 Systemrettung nach Löschen von `/var` . . . 74
6.3.6 Installation eines Pakets in ein nicht bootfähiges System . . . 75

6.3.7 Der dpkg Befehl funktioniert nicht mehr – was nun? . . . 75
6.4 Debian Nirvana Befehle . . . 76
6.4.1 Informationen über eine Datei . . . 76
6.4.2 Informationen über ein Paket . . . 76
6.4.3 Nicht-Interaktive Installation mit APT . . . 77
6.4.4 Installierte Pakete erneut konfigurieren . . . 77
6.4.5 Deinstallation von Paketen . . . 78
6.4.6 "Halten" älterer Pakete . . . 78
6.4.7 Gemischtes `stable/testing/unstable` System . . . 79
6.4.8 Löschen von zwischengespeicherten Paketdateien . . . 79
6.4.9 Speichern/Kopieren der Systemkonfiguration . . . 80
6.4.10 Portierung eines Pakets auf die `stable` Distribution . . . 80
6.4.11 Lokale Paketarchive . . . 81
6.4.12 Installation und Konvertierung eines fremden Binärpakets . . . 82
6.4.13 Verifizierung installierter Paketdateien . . . 82
6.4.14 Optimierte `sources.list` . . . 82
6.5 Weitere Debian Besonderheiten . . . 83
6.5.1 Der `dpkg-divert` Befehl . . . 83
6.5.2 Das `equivs` Paket . . . 83
6.5.3 Alternative Befehle . . . 83
6.5.4 System-V `init` und Runlevel . . . 84
6.5.5 Deaktivierung von Daemon Diensten . . . 85

7 Der Linux Kernel unter Debian **87**
7.1 Neukompilieren des Kernels . . . 87
7.1.1 Die Debian Standard Methode . . . 88
7.1.2 Klassische Methode . . . 88
7.1.3 Kernelheader . . . 89
7.2 Der modularisierte Kernel 2.4 . . . 89
7.2.1 PCMCIA . . . 90
7.2.2 SCSI . . . 90

7.2.3 Netzwerk Funktionalität . . . 91
7.2.4 EXT3 Dateisystem (> 2.4.17) . . . 92
7.2.5 Realtek RTL-8139 Unterstützung in Kernel 2.4 . . . 93
7.2.6 Parallel Port Unterstützung . . . 93
7.3 Tunen des Kernels mittels des proc Dateisystems . . . 93
7.3.1 Zu viele geöffnete Dateien . . . 94
7.3.2 Disk Flush Intervalle . . . 94
7.3.3 Trägheit alter Maschinen mit wenig Speicher . . . 94

8 Debian Tipps **95**
8.1 Booten des Systems . . . 95
8.1.1 "Ich habe das root-Passwort vergessen!" (1) . . . 95
8.1.2 "Ich habe das root-Passwort vergessen!" (2) . . . 96
8.1.3 Das System kann nicht gestartet werden . . . 96
8.1.4 "Let me disable X on boot!" . . . 97
8.1.5 Weitere Tricks mit dem Boot-Prompt . . . 98
8.1.6 How do I set boot parameters (GRUB) . . . 98
8.2 Aufzeichnung und Abspielen von Aktivitäten . . . 99
8.2.1 Shell-Eingaben . . . 99
8.2.2 X Aktivitäten aufzeichnen . . . 99
8.3 Kopieren und Archivieren eines Verzeichnisbaumes . . . 100
8.3.1 Grundlegende Kommandos zum Kopieren eines Unterverzeichnisses . . . 100
8.3.2 `cp` . . . 100
8.3.3 `tar` . . . 101
8.3.4 `pax` . . . 101
8.3.5 `cpio` . . . 101
8.3.6 `afio` . . . 101
8.4 Differential backup and data synchronization . . . 102
8.4.1 Differential backup with rdiff . . . 102
8.4.2 Regular differential backup with RCS . . . 103
8.5 Einfrieren von Prozessen . . . 103

8.5.1 Einen Prozess killen/beenden . . . 103
8.5.2 ALT-SysRq . . . 103
8.6 Raffinierte merk-würdige Kommandos . . . 104
8.6.1 Dateibetrachter . . . 104
8.6.2 Freier Speicher . . . 104
8.6.3 Setzen der Uhrzeit (BIOS) . . . 104
8.6.4 Setzen der Uhrzeit (NTP) . . . 105
8.6.5 Den Bildschirmschoner deaktivieren . . . 105
8.6.6 Durchsuchen von Datenbanken zur Systemverwaltung . . . 106
8.6.7 Sound abstellen (beep) . . . 106
8.6.8 Fehlermeldungen auf der Konsole . . . 106
8.6.9 Setzen des korrekten Konsole-Typs . . . 106
8.6.10 Die Konsole wiederherstellen . . . 107
8.6.11 Konvertieren einer Textdatei von DOS nach Unix . . . 107
8.6.12 Reguläre Ausdrücke . . . 107
8.6.13 Extract differences and merging updates for the source file . . . 108
8.6.14 Eine große Datei in kleine zerlegen . . . 108
8.6.15 Kleine Tricks mit Pipes . . . 108
8.6.16 Text oder Mailinglist aus einer Webseite extrahieren . . . 109
8.6.17 Formatierte Ausgabe von Webseiten . . . 109
8.6.18 Pretty print a manual page . . . 109
8.6.19 Merge two Postscript or PDF files . . . 109
8.6.20 Ausführungsdauer messen . . . 110
8.6.21 `nice`-Befehl . . . 110
8.6.22 Terminplanung für Prozesse (`cron`, `at`) . . . 110
8.6.23 Konsolenumschaltung mit `screen` . . . 111
8.6.24 Grundlagen - Prüfung des Netzwerks . . . 112
8.6.25 Email aus dem lokalen Spooler leiten (flush) . . . 113
8.6.26 'Frozen' Email aus dem lokalen Spooler entfernen . . . 113
8.6.27 Re-deliver mbox contents . . . 113
8.6.28 Dateiinhalte zurücksetzen . . . 113

8.6.29 Dummy-Dateien . . . 114
8.6.30 `chroot` . . . 114
8.6.31 How to check hard links . . . 114
8.6.32 Ein Festplatten-Image `mounten` . . . 115
8.6.33 Samba . . . 115

9 Tunen eines Debian-Systems **117**
9.1 Hinweise zur System-Inititalisierung . . . 117
9.1.1 Anpassen der Init-Skripte . . . 117
9.1.2 Anpassen der System-Protokollierung . . . 117
9.1.3 Hardware-Optimierung . . . 118
9.2 Zugriffskontrolle . . . 119
9.2.1 Zugriffskontrolle mit PAM und login . . . 119
9.2.2 "Warum GNU `su` nicht die `wheel` Gruppe unterstützt" . . . 120
9.2.3 Bedeutung verschiedener Gruppen . . . 120
9.2.4 `sudo` – eine sicherere Arbeitsumgebung . . . 121
9.2.5 Zugriffskontrolle auf Daemonen . . . 121
9.2.6 LDAP – Leichtgewichtiges Verzeichniszugangsprotokoll . . . 122
9.3 CD-Brenner . . . 122
9.3.1 Einführung . . . 123
9.3.2 Ansatz 1: Module + `lilo` . . . 123
9.3.3 Ansatz 2: Neukompilieren des Kernels . . . 123
9.3.4 Weitere Schritte nach der Konfiguration . . . 124
9.3.5 Bootfähiges CD-Image . . . 125
9.3.6 Brennen mit dem CD-Brenner (R, RW): . . . 125
9.3.7 Eine Abbilddatei (Image) einer CD erzeugen . . . 126
9.3.8 Debian-CD-Images . . . 126
9.3.9 Sicherheitskopien auf CD-R . . . 127
9.3.10 Eine Musik-CD kopieren . . . 127
9.4 Das X-Window Paket . . . 127
9.4.1 X-System Pakete . . . 128

9.4.2 Hardware-Erkennung für X . . . 129
9.4.3 X-Server . . . 129
9.4.4 X-Client Programme . . . 131
9.4.5 X-Session . . . 132
9.4.6 TCP/IP-Verbindungen mit X . . . 136
9.4.7 X-Fernverbindungen mit `xhost` . . . 136
9.4.8 X-Fernverbindungen mit `ssh` . . . 136
9.4.9 `xterm` . . . 137
9.4.10 X-Ressourcen Datenbank . . . 137
9.4.11 Administratorrechte unter X . . . 138
9.4.12 TrueType-Schriften unter X . . . 139
9.4.13 Web-Browser (graphische) . . . 140
9.5 SSH . . . 140
9.5.1 Grundlagen . . . 140
9.5.2 Port-Umleitung – für das Tunneln mit SMTP/POP3 . . . 142
9.5.3 Mit weniger Passwörter agieren . . . 142
9.5.4 Weitere SSH-Client-Programme . . . 143
9.5.5 SSH-Agent . . . 143
9.5.6 Problemlösungen . . . 144
9.6 E-Mail Programme . . . 144
9.6.1 E-Mail Transport/Transfer Agenten (Exim) . . . 144
9.6.2 Mail-Hilfsprogramme (Fetchmail) . . . 145
9.6.3 Mail-Hilfsprogramme (Procmail) . . . 146
9.6.4 Email-Programme (Mail user agent) am Beispiel Mutt . . . 146
9.7 Lokalisation und Sprachen . . . 147
9.7.1 Grundlagen . . . 147
9.7.2 Lokalisierung mit locale . . . 148
9.7.3 Aktivierung der Lokalisierung . . . 148
9.7.4 Eine Locale aktivieren . . . 148
9.7.5 ISO 8601 Datum-Format-locale . . . 149
9.7.6 Beispiel für die USA (ISO-8859-1) . . . 149

9.7.7 Beispiel für Frankreich mit Euro-Symbol (ISO-8859-15) . . . 150
9.7.8 Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1) 150
9.7.9 Beispiel für UTF-8 in X . . . 152
9.7.10 Beispiel für UTF-8 in der FB-Konsole . . . 152
9.7.11 Über locale hinaus . . . 152

10 Aufsetzen eines Gateway mit einem Debian System 155
10.1 Netzwerkkonfiguration . . . 155
10.1.1 Rechnerkonfiguration des Gateway . . . 155
10.1.2 Anhaltspunkte zur Netzwerkkonfiguration . . . 156
10.2 Netfilter Konfiguration . . . 157
10.2.1 Grundlagen von netfilter . . . 157
10.2.2 Netfilter Tabelle . . . 158
10.2.3 Netfilter Ziele . . . 158
10.2.4 Netfilter Kommandos . . . 159
10.2.5 IP-Masquerading . . . 159
10.2.6 Umleiten von SMTP Verbindungen (2.4) . . . 160
10.3 Handhaben verschiedener Netzverbindungen . . . 160

11 Editoren 161
11.1 Populäre Editoren . . . 161
11.2 Rettungseditoren . . . 161
11.3 Emacs und Vim . . . 162
11.3.1 Vim Hinweise . . . 162
11.3.2 Emacs Hinweise . . . 162
11.3.3 Starten des Editors . . . 162
11.3.4 Zusammenfassung der Editorkommandos (Emacs, Vim) . . . 163
11.3.5 Vim Konfiguration . . . 165
11.3.6 Ctags . . . 165
11.3.7 Konvertierung eines mit Syntax-Highlighting versehenen Schirms in HTML . . . 165
11.3.8 Aufteilen des Bildschirms mit `vim` . . . 166

12 Versions-Kontroll-Systeme **167**

12.1 CVS . . . 167

12.1.1 Installation eines CVS Servers . . . 167

12.1.2 Beispiele für CVS Sitzungen . . . 167

12.1.3 Fehlersuche im CVS . . . 171

12.1.4 CVS Kommandos . . . 171

12.2 Subversion . . . 172

12.2.1 Installation eines Subversion Servers . . . 172

12.2.2 Verschieben eines CVS Repositories nach Subversion . . . 173

12.2.3 Anwendungsbeispiele von Subversion . . . 173

13 Programmierung **175**

13.1 Wo man startet . . . 175

13.2 Shell . . . 175

13.2.1 Bash – interaktive **GNU** Standard Shell . . . 175

13.2.2 POSIX Shells . . . 176

13.2.3 Shellparameter . . . 177

13.2.4 Shellumleitung . . . 177

13.2.5 Bedingte Ausdrucke in der Shell . . . 178

13.2.6 Kommandozeilenbearbeitung . . . 179

13.3 Awk . . . 179

13.4 Perl . . . 181

13.5 Python . . . 181

13.6 Make . . . 183

13.7 C . . . 184

13.7.1 Ein einfaches C Programm (`gcc`) . . . 184

13.7.2 Fehlersuche . . . 185

13.7.3 Flex – ein besseres Lex . . . 187

13.7.4 Bison – ein besseres Yacc . . . 187

13.7.5 Autoconf . . . 188

13.8 Dokument Aufbereitung . . . 188

13.8.1 Das `roff` Satzsystem . . . 188
13.8.2 SGML . . . 189
13.8.3 TeX/LaTeX . . . 190
13.8.4 Literate Programming . . . 191
13.9 Paketerzeugung . . . 193
13.9.1 Paketerzeugung für ein einzelnes Programm . . . 193
13.9.2 Paketerzeugung mit Tools . . . 193

14 GnuPG **195**
14.1 Installation von GnuPG . . . 195
14.2 Verwendung von GnuPG . . . 196
14.3 Umgang mit GnuPG . . . 196
14.4 Benutzung von GnuPG mit Anwendungen . . . 197
14.4.1 Benutzung von GnuPG mit Mutt . . . 197
14.4.2 Benutzung von GnuPG mit Vim . . . 197

15 Unterstützung für Debian **199**
15.1 Referenzen . . . 199
15.2 Finden der Bedeutung eines Wortes . . . 203
15.3 Bestimmen der Popularität eines Debian Pakets . . . 203
15.4 Das Debian Fehlerverfolgungssystem . . . 203
15.5 Mailing-Listen . . . 204
15.6 Internet Relay Chat (IRC) . . . 204
15.7 Suchmaschinen . . . 204
15.8 Webseiten . . . 205

A Anhang **207**
A.1 Autoren . . . 207
A.2 Haftung . . . 209
A.3 Rückmeldungen . . . 209
A.4 Dokumentformat . . . 210
A.5 Das Debian Labyrinth . . . 210
A.6 Debian Zitate . . . 211

Kapitel 1

Vorwort

Diese Debian Referenz (http://qref.sourceforge.net/) wurde geschrieben, um einen groben Überblick über das Debian System in Form einer **Referenz für die Zeit nach der Installation** zu gewinnen. Es spricht die Leser an, welche bereit sind, Shell Skripte zu lesen. Ich erwarte, dass der Leser bereits hinreichende Erfahrungen mit Unix-artigen Systemen vor dem Lesen dieses Dokuments gesammelt hat.

Ich traf die bewusste Entscheidung **nicht** alles im Detail zu erklären, wenn es in einer **Manual-Page**, einer **info Seite** oder einem **HOWTO Dokument** gefunden werden kann. Anstatt alles ausführlich zu erklären, habe ich mehr Wert auf direkte praktische Informationen durch Angabe **exakter Kommandosequenzen** im Text oder **Beispielskripte** unter examples/ als Referenz gelegt. Sie müssen den Inhalt der Beispiele verstanden haben, bevor Sie sie ausführen. Möglicherweise müssen Sie die Kommandosequenzen auch leicht abändern.

Viele der aufgeführten Informationen entstammen meinem Gedächtnis oder Verweisen auf die ultimativen Referenzen, welche unter 'Referenzen' auf Seite 199 aufgelistet sind. Dies geschah teilweise, weil dieses Dokument ursprünglich als eine "**Schnellreferenz**" gedacht war.

Halte es kurz und einfach ist mein Leitmotiv.

Für Hilfe im Falle von Notfall Wartung wird auf 'Debian Überlebensbefehle' auf Seite 71 verwiesen.

1.1 Das offizielle Dokument

Das neueste offizielle Dokument ist im Debian Archiv im Paket debian-reference und ebenso unter http://www.debian.org/doc/manuals/debian-reference/ verfügbar.

Die aktuellste Entwicklerversion ist http://qref.sourceforge.net/Debian/. Das Projekt wird auf http://qref.sourceforge.net/ gehostet, und dieses Dokument ist da zum Herunterladen als einfacher Text und im HTML, PDF, SGML sowie PostScript Format verfügbar.

1.2 Konventionen in diesem Dokument

Diese “Debian Referenz” enthält Informationen in Form von kurzen Bash Shell Kommandos. Die folgenden Notationen werden verwendet:

```
# Kommando im root account
$ Kommando im Nutzer account
... Beschreibung der Auswirkungen
```

Vergleiche ‘Bash – interaktive **GNU** Standard Shell’ auf Seite 175 für weitere Informationen zur Bash.

Verweise auf:

- eine **Unix Manual-Page** erfolgen in der Form `bash(1)`.
- eine **GNU TEXINFO** Seite erfolgen in der Form `info libc`.
- ein **Buch** erfolgen in der Form *Die Programmiersprache C*.
- eine **URL** erfolgen in der Form `http://www.debian.org/doc/manuals/debian-reference/`.
- eine **Datei** im System erfolgen in der Form `/usr/share/doc/Debian/reference/`.

Die folgenden Abkürzungen werden verwendet:

- **LDP**: Linux Dokumentations Projekt (`http://www.tldp.org/`)
- **DDP**: Debian Dokumentations Projekt (`http://www.debian.org/doc/`)

In diesem Dokument wird nur auf URLs von LDP Dokumenten verwiesen, sie können aber auch als Paket erhalten und unter `/usr/share/doc/HOWTO/` installiert werden. Vergleiche ‘Referenzen’ auf Seite 199.

1.3 Beispielskripte

Beispielskripte sind im Beispielverzeichnis (`examples/`), installiert unter `/usr/share/doc/Debian/reference/` verfügbar; bei versteckten Dateien wird der führende Punkt “.” im Dateinamen zu einem Unterstrich “_” konvertiert. Spezielle Erweiterungen werden verwendet, wenn es mehrere alternative Beispielskripte gibt.

1.4 Grundlegende Einstellungen

Ist das System mit dem absoluten Minimum an Paketen installiert, so sollten die folgenden Kommandos ausgeführt werden, um einige wichtige Pakete und Schlüsseldokumente zu installieren:

```
# apt-get install info man-db doc-base dhelp apt apt-utils auto-apt \
                  dpkg less mc ssh nano-tiny elvis-tiny vim sash \
```

```
                    kernel-package \
                    manpages manpages-dev doc-debian doc-linux-text \
                    debian-policy developers-reference maint-guide \
                    apt-howto harden-doc install-doc \
                    libpam-doc glibc-doc samba-doc exim-doc cvsbook \
                    gnupg-doc
  # apt-get install debian-reference # für Sarge unbedingt zu empfehlen
```

1.5 Grundlagen der Debian Distributionen

Debian ist in drei Ausgaben verfügbar:

- `stable`: Gut für einen wichtigen Server aber langweilig für eine Workstation. Vergleiche 'Die `stable` Distribution' auf Seite 6.
- `testing`: Gut zum Aufspielen auf eine Workstation geeignet. Vergleiche 'Die `testing` Distribution' auf Seite 7.
- `unstable`: Sollte niemals blind installiert werden. Vergleiche 'Die `unstable` Distribution' auf Seite 7.

Zumindest die Mailing Liste `debian-devel-announce@lists.debian.org` sollte wegen der Informationen über Updates für Debian gelesen werden.

Im März 2002 entsprachen diese drei Ausgaben den Kodenamen-Versionen Potato (Produktionsqualität), Woody (Beta-Test, sehr stabil, da die Veröffentlichung bevorstand) und Sid (Alpha-Test). Im August 2002, kurz nach der Woody Ausgabe, entsprachen diese Woody (Produktionsqualität), Sarge (Beta-Test, wird manchmal etwas eigenwillig sein) und Sid (ständig Alpha-Test). Tauchen zu Paketen in `unstable` keine ausgabekritischen Fehler innerhalb der ersten Woche (ca.) auf, so werden sie automatisch `testing` zugeordnet. Vergleiche 'Die Debian Archive' auf Seite 5.

Es gibt zwei Möglichkeiten, um die neueste Software auf einem System zu verwenden.

- 'Aktualisieren einer Distribution auf `testing`' auf Seite 61 (hauptsächlich für Workstations)
- 'Portierung eines Pakets auf die `stable` Distribution' auf Seite 80 (hauptsächlich für Server)

Das Verwenden der `testing` Ausgabe von Debian hat den Nebeneffekt, das Sicherheitsupdates nur langsam eingespielt werden. Seien Sie gewarnt.

Vermengen Sie verschiedene Ausgaben von Debian, indem Sie Teile von `testing` oder `unstable` in `stable` verwenden, so werden wahrscheinlich unabsichtlich viele Schlüsselpakete von `testing` oder `unstable` verwendet, die fehlerhaft sein können. Seien Sie gewarnt.

Die Verwendung der `testing` oder `unstable` Ausgaben von Debian führt zum erhöhten Risiko, auf verschiedene Fehler zu stoßen. Dieses Risiko kann durch ein Multi-Boot Schema

gemanagt werden, bei dem eine stabilere Ausgabe von Debian genutzt wird. Alternativ kann auch ein netter Trick mit `chroot` und der stabileren Version wie in ‘`chroot`’ auf Seite 114 beschrieben, genutzt werden. Letzteres ermöglicht die Nutzung verschiedener Versionen von Debian simultan auf verschiedenen Konsolen.

Nach dem Erklären einiger Grundlagen der Debian Distribution in ‘Debian Grundlagen’ auf der nächsten Seite, werde ich einige wichtige Informationen für die Verwendung der aktuellsten Software, unter Ausnutzung der Vorteile der `testing` und `unstable` Distributionen von Debian, präsentieren. Die Ungeduldigen sollten sofort mit ‘Debian Überlebensbefehle’ auf Seite 71 fortfahren. Glückliches upgraden!

Kapitel 2

Debian Grundlagen

Dieses Kapitel liefert grundlegende Informationen über das Debian System für Nicht-Programmierer. Für die ultimativen Informationen vergleiche:

- Debian Policy Manual
- Debian Packaging Manual (Potato)
- Debian Developer's Reference
- Debian New Maintainers' Guide

aufgeführt unter 'Referenzen' auf Seite 199.

Wenn Sie nach nicht allzu ausführlichen "wie geht" Erklärungen suchen, wechseln Sie direkt nach 'Debian Paketverwaltung' auf Seite 65 oder in andere relevante Kapitel.

Dieses Kapitel basiert auf Dokumenten aus der "Debian FAQ", größtenteils umgeschrieben, um dem gewöhnlichen Debian Systemadministrator den Einstieg zu erleichtern.

2.1 Die Debian Archive

2.1.1 Verzeichnisstrukturen

Die Software, welche für Debian gepackt wurde, ist in einem der vielen Verzeichnisbäume auf jedem Debian Mirror (`http://www.debian.org/misc/README.mirrors`) durch FTP oder HTTP verfügbar.

Die folgenden Verzeichnisse können auf jedem Debian Mirror unter dem `debian` Verzeichnis gefunden werden:

`dists/`: Dieses Verzeichnis enthält die "Distributionen" und ist für den Zugriff auf die aktuell verfügbaren Pakete in Debian ausgelegt. Einige alte Pakete und die `Packages.gz` Dateien sind immer noch hier zu finden.

`pool/`: Die neue Position aller Debian Pakete.

tools/: DOS Hilfsmittel zum Erzeugen von Bootdisketten, Partitionieren der Festplatte, Komprimieren/Dekomprimieren von Dateien und zum Booten von Linux.

doc/: Die grundlegenden Debian Dokumentationen wie die FAQ's, Erläuterungen zum Fehler-Melde-System, usw.

indices/: Enthält die `Maintainers` Datei (Liste aller Paketbetreuer) und die `override.*` Dateien.

project/: Hauptsächlich Material, welches nur für Entwickler von Interesse ist, wie z.B.:

project/experimental/: Dieses Verzeichnis enthält Pakete und Hilfsmittel, welche noch entwickelt werden und sich noch im Alpha-Stadium befinden. Benutzer sollten keine Pakete von hier verwenden, da sie selbst für den Erfahrensten gefährlich und schädlich sind.

project/orphaned/: Pakete, welche von ihrem alten Betreuer aufgegeben wurden und aus der Distribution zurückgezogene Software.

2.1.2 Debian Distributionen

Normalerweise befinden sich drei Debian Distributionen im `dists` Verzeichnis. Dies sind die `stable`, die `testing` und die `unstable` Distribution. Manchmal gibt es auch eine `frozen` Distribution. Jede Distribution ist ein symbolischer Link in das entsprechende Verzeichnis mit einem Kodenamen im `dists` Verzeichnis.

2.1.3 Die `stable` Distribution

Pakete der `stable` Distribution, Debian Woody (3.0r0), befinden sich im `stable` (symbolischer Link zu `woody/`) Verzeichnis:

- `stable/main/`: Dieses Verzeichnis enthält die Pakete, welche formal die aktuellste Ausgabe des Debian Systems bilden.

  Diese Pakete entsprechen alle den Debian Free Software Leitlinien (DFSG) (`http://www.debian.org/social_contract#guidelines`) (auch verfügbar unter `/usr/share/doc/debian/social-contract.txt` installiert durch `debian-doc`) und sind alle frei verfügbar und verteilbar.

- `stable/non-free/`: Dieses Verzeichnis enthält Pakete, deren Weitergabe in gewisser Weise beschränkt ist. Genauere Informationen sind in den jeweiligen Copyright Hinweisen zu finden.

  Zum Beispiel verbieten die Lizenzen einiger Pakete die kommerzielle Verteilung. Andere können weitergegeben werden, aber nur als Shareware und nicht als freie Software. Die Lizenzen all dieser Pakete müssen genau studiert und möglicherweise ausgehandelt werden, bevor die Pakete in irgendeiner Form (z.B. auf einer CD-ROM) weitergegeben werden.

- `stable/contrib/`: Dieses Verzeichnis enthält Pakete, welche DFSG-frei und **frei verteilbar** sind, aber irgendwie von einem Paket abhängen, das **nicht** frei verteilbar und somit nur im non-free Abschnitt zu finden ist.

Gegenwärtig befinden sich neue Pakete in Ergänzung zu obigen Verzeichnissen unterhalb des `pool` Verzeichnisses ('Das `pool` Verzeichnis' auf Seite 9).

Der aktuelle Status von Fehlern der `stable` Distribution ist unter der Stable Problems (`http://ftp-master.debian.org/testing/stable_probs.html`) Web Seite aufgeführt.

2.1.4 Die `testing` Distribution

Pakete der `testing` Distribution, Debian Sarge, befinden sich im `testing` (symbolischer Link zu `sarge/`) Verzeichnis, nachdem sie einige Zeit in `unstable` getestet wurden. Gegenwärtig befinden sich neue Pakete, im Gegensatz zu den obigen Positionen, unterhalb des `pool` Verzeichnisses ('Das `pool` Verzeichnis' auf Seite 9). Auch in `testing/` gibt es die `main`, `contrib` und `non-free` Unterverzeichnisse, diese entsprechen den Verzeichnissen in `stable/`.

Diese Pakete müssen auf allen Architekturen auf denen sie zur Verfügung stehen gleich aktuell sein und dürfen keine Abhängigkeiten aufweisen, welche sie nicht installierbar machen; sie müssen auch weniger ausgabekritische Fehler haben, als die Versionen in `unstable`. Auf diese Art hoffen wir, dass `testing` fast immer zur Veröffentlichung bereit ist. Mehr Details zu den Test Mechanismen sind unter `http://ftp-master.debian.org/testing/` verfügbar.

Der aktuellste Status der `testing` Distribution ist unter folgenden Seiten aufgeführt:

- update Ausflüchte (`http://ftp-master.debian.org/testing/update_excuses.html`)
- testing Probleme (`http://ftp-master.debian.org/testing/testing_probs.html`)
- ausgabekritische Fehler (`http://bugs.debian.org/release-critical/`)
- Basissystem Fehler (`http://bugs.qa.debian.org/cgi-bin/base.cgi`)
- Fehler in Standard und Task Paketen (`http://bugs.qa.debian.org/cgi-bin/standard.cgi`)
- andere Fehler und Bemerkungen zu Fehlerausmerzungs Parties (`http://bugs.qa.debian.org/`)

2.1.5 Die `unstable` Distribution

Pakete, welche zur `unstable` Distribution mit dem Kodenamen "Sid" gehören, werden im `unstable` (symbolischer Link zu `sid/`) Verzeichnis aufbewahrt, nachdem sie in das Debian Archiv geladen wurden und verbleiben hier solange, bis sie nach `testing/` verschoben werden. Neue Pakete befinden sich unterhalb des `pool` Verzeichnis ('Das `pool` Verzeichnis' auf Seite 9). Es gibt auch `main`, `contrib` und `non-free` Unterverzeichnisse in `unstable/`, welche dem selben Zweck dienen wie in `stable/`.

Die `unstable` Distribution enthält ein Abbild des aktuellsten Entwickler Systems. Benutzer sind willkommen diese Pakete zu benutzen und zu testen, werden aber über den Status der Einsatzbereitschaft gewarnt. Der Vorteil der Benutzung der `unstable` Distribution ist, dass man immer auf dem Laufenden mit der aktuellsten Debian Software ist – geht jedoch etwas schief, so muss man auch damit umgehen können. :-)

Der aktuelle Status der Fehler in der `unstable` Distribution wird auf der Unstable Problems (`http://ftp-master.debian.org/testing/unstable_probs.html`) Web Seite aufgeführt.

2.1.6 Die `frozen` Distribution

Wenn die `testing` Distribution ausgereift ist, so wird aus ihr frozen, was bedeutet, dass kein neuer Code mehr akzeptiert wird, nur noch Bugfixes, wenn nötig. Es wird auch ein neuer Verzeichnisbaum im `dists` Verzeichnis angelegt und einem neuen Kodenamen zugeordnet. Die eingefrorene Distribution durchläuft nun einige Monate lang Tests mit zwischenzeitlichen Updates und Zwischenausgaben, welche "Test Zyklen" genannt werden. (Der aktuelle Woody Ausgabeprozess erzeugte keinen symbolischen Link `frozen/`, deshalb war `frozen` keine Distribution sondern nur ein Entwicklungsschritt der `testing` Distribution.)

Wir führen eine Liste aller Fehler in der frozen Distribution, welche die Veröffentlichung eines Paketes verzögern können, sowie von Fehlern, welche ähnliche Auswirkungen auf die gesamte Ausgabe haben. Sobald die Anzahl der Fehler den maximal zulässigen Wert unterschreitet, wird aus der eingefrorenen Distribution stable, sie wird veröffentlicht und die letzte stable Distribution veraltet (und wird ins Archiv verschoben).

2.1.7 Debian Distributions-Kodenamen

Verzeichnisnamen im `dists` Verzeichnis, wie `woody/` und `sarge/` sind nur "Kodenamen". Wenn sich eine Debian Distribution in der Entwicklung befindet, besitzt sie keine Versionsnummer sondern nur einen Kodenamen. Der Grund für diese Kodenamen ist das Spiegeln der Debian Distributionen zu vereinfachen (wenn ein Verzeichnis wie `unstable` plötzlich zu `stable/` umbenannt wird, so müsste vieles erneut heruntergeladen werden).

Zur Zeit ist `stable/` ein symbolischer Link zu `woody/` und `testing/` ist ein symbolischer Link zu `sarge/`. Das bedeutet, dass `Woody` die aktuelle stable Distribution und `Sarge` die aktuelle testing Distribution ist.

`unstable/` ist ein permanenter symbolischer Link zu `sid/`, so wie Sid ständig für die unstable Distribution steht.

2.1.8 In der Vergangenheit verwendete Kodenamen

Andere bereits verwendete Kodenamen sind: "Buzz" für Ausgabe 1.1, "Rex" für Ausgabe 1.2, "Bo" für die Ausgaben 1.3.x, "Hamm" für Ausgabe 2.0, "Slink" für Ausgabe 2.1 und "Potato" für Ausgabe 2.2.

2.1.9 Die Herkunft der Kodenamen

Bisher wurden Personen aus dem Film *Toy Story* von Pixar verwendet.

- **Buzz** (Buzz Lightyear) war der Astronaut,
- **Rex** war der Tyrannosaurus,
- **Bo** (Bo Peep) war das Mädchen, das sich um die Schafe kümmerte,
- **Hamm** war das Sparschwein,
- **Slink** (Slinky Dog) war der Spielzeughund,
- **Sarge** was der Anführer der grünen Plastik Armee Männer,
- **Potato** war natürlich Mr. Potato Head (Mr. Kartoffelkopf),
- **Woody** war der Cowboy,
- **Sid** war ein Nachbarsjunge welcher Spielzeug zerstörte.

2.1.10 Das `pool` Verzeichnis

Früher wurden Pakete in dem Unterverzeichnis von `dists` aufbewahrt, welches der verwendeten Distribution entsprach. Es stellte sich heraus, dass dies einige Probleme verursachte, wie z.B. große Bandbreitenverschwendung auf Mirrors nach einigen großen Änderungen.

Pakete werden nun in einem großen "Pool" gespeichert, entsprechend dem Namen des Quellpakets strukturiert. Um dies handhaben zu können, wurde der Pool je nach Abschnitt (`main`, `contrib` und `non-free`) sowie dem ersten Buchstaben des Quellpakets unterteilt. Diese Verzeichnisse enthalten verschiedene Dateien: die Binärpakete für jede Architektur und das Quellpaket von welchem die Binärpakete erzeugt wurden.

Man kann herausfinden wo sich ein Paket befindet, indem man ein Kommando wie `apt-cache showsrc` *`Paketname`* aufruft und nach der "Directory:" Zeile schaut. Das `apache` Paket wird z.B. unter `pool/main/a/apache/` gespeichert. Da es sehr viele `lib*` Pakete gibt, werden diese gesondert behandelt: das `libpaper` Paket wird beispielsweise unter `pool/main/libp/libpaper/` gespeichert.

Die `dists` Verzeichnisse werden nach wie vor für die Index Dateien, welche von Programmen wie `apt` verwendet werden, genutzt. Ebenso wurden während dies geschrieben wird, ältere Distributionen noch nicht angepasst um Pools zu nutzen, deswegen werden Sie auch Pfade finden, die den Distributionsnamen wie `potato` oder `woody` im "Directory" Feld enthalten.

Normalerweise muss man sich um dies nicht kümmern, da neue `apt` und wahrscheinlich ältere `dpkg-ftp` Programme (vergleiche 'Methoden zum Aktualisieren eines Debian Systems' auf Seite 20) dies problemlos handhaben. Sind Sie an weiteren Informationen interessiert, so sei auf die RFC: Implementation von Paketpools (`http://lists.debian.org/debian-devel-announce/2000/debian-devel-announce-200010/msg00007.html`) verwiesen.

2.1.11 Historische Bemerkungen über Sid

Als das heutige Sid noch nicht existierte, gab es im Debianarchiv nur einen Zweig für nicht ausgereifte Pakete: es gab die Annahme, dass, wenn eine Architektur im aktuellen `unstable`

/ hinzukam, sie veröffentlicht wurde, wenn diese Distribution zum neuen `stable` Zweig wurde. Für viele Architekturen war das nicht der Fall, was dazu führte, dass diese Verzeichnisse während der Veröffentlichung verschoben wurden. Dies war unpraktisch, da die Verschiebung zu einer großen Bandbreitenbelastung führte.

Die Archiv Administratoren umgingen das Problem einige Jahre, indem sie Binaries für nichtveröffentlichte Architekturen in einem speziellen Verzeichnis namens `sid` bereitstellten. Für solche noch nicht veröffentlichte Architekturen wurde das erste Mal als sie veröffentlicht wurden ein Link vom aktuellen `stable/` zu `sid/` angelegt, und später wurden sie wie üblich unter `unstable/` veröffentlicht. Diese Vorgehensweise war zum Teil für die Anwender verworren.

Mit Beginn der Paket Pools (vergleiche 'Das `pool` Verzeichnis' auf der vorherigen Seite) während der Entwicklung der Woody Distribution, wurden Binärpakete unabhängig von der Distribution vorschriftsmäßig im Pool gehalten, so dass die Veröffentlichung einer Distribution nicht länger zu einer großen Bandbreitenverschwendung auf den Mirrors führte (es gibt dennoch während der Entwicklung eine relativ große Bandbreitenauslastung).

2.1.12 Heraufgeladene Pakete in `incoming/`

Heraufgeladene Pakete befinden sich zunächst unter `http://incoming.debian.org/` nachdem sie überprüft wurden, um sicherzustellen, dass sie wirklich von einem Debian Entwickler stammen, (und sie werden in das `DELAYED` Unterverzeichnis verschoben, wenn das Paket nicht von einem Entwickler stammt, d.h. ein Non-Maintainer Upload (NMU) ist). Einmal pro Tag werden sie von `incoming/` nach `unstable/` verschoben.

Im Notfall kann man Pakete aus `incoming/` installieren, bevor sie nach `unstable/` kommen.

2.1.13 Wiederauffinden eines älteren Paketes

Während die aktuellsten Debian Distributionen unter dem `debian` Verzeichnis auf jedem Debian Mirror (`http://www.debian.org/misc/README.mirrors`) zu finden sind, werden die Archive für ältere Debian Distributionen wie Slink unter `http://archive.debian.org/` oder unterhalb des `debian-archive` Verzeichnis auf jedem Debian Mirror gehalten.

Ältere `testing` und `unstable` Pakete befinden sich auf `http://snapshot.debian.net/`.

2.1.14 Architektur abhängige Verzeichnisse

Innerhalb jedes der wichtigen Verzeichnisbäume (`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free`, `dists/unstable/main/`, etc.), befinden sich die Einträge der Binärpakete in Unterverzeichnissen, deren Namen die Prozessorarchitektur kennzeichnen, für die sie kompiliert wurden.

- `binary-all/` für Pakete, welche Architektur unabhängig sind. Dies umschließt z.B. Perl Skripte oder Dokumentationen.
- `binary-`*`Plattform`*`/` für Pakete, welche sich auf einer einzelnen Binärplattform starten lassen.

Es ist anzumerken, dass die aktuellen Binärpakete von `testing` und `unstable` sich nicht mehr länger in diesen Verzeichnissen, dafür aber im `pool` Verzeichnis befinden. Die Index Dateien (`Packages` und `Packages.gz`) befinden sich nach wie vor für Abwärtskompatibilität dort.

Für die aktuell unterstützten Binärarchitekturen vergleiche die Release Notes der einzelnen Distributionen. Sie können unter der Release Notes Seite für stable (`http://www.debian.org/releases/stable/releasenotes`) und testing (`http://www.debian.org/releases/testing/releasenotes`) gefunden werden.

2.1.15 Der Quellcode

Der Quellcode für alles im Debian System ist mit darin enthalten. Außerdem **fordern** die Lizenzvereinbarungen der meisten Programme im System, dass der Quellcode zusammen mit den Programmen ausgeliefert wird, oder dass dem Programm Informationen beiliegen, wie er zu erhalten ist.

Normalerweise befindet sich der Quellcode in den `source` Verzeichnissen, welche parallel zu allen Architektur spezifischen Binärverzeichnissen sind oder aktueller im `pool` Verzeichnis (vergleiche 'Das `pool` Verzeichnis' auf Seite 9). Um den Quellcode zu erhalten, ohne sich mit der Verzeichnisstruktur des Debian Archivs auseinandersetzen zu müssen, kann ein Befehl wie `apt-get source `*`Meinpaketname`* genutzt werden.

Einige Pakete wie z.B. `pine`, sind wegen deren Lizenzbedingungen nur im Quellcode erhältlich. (Kürzlich wurde das `pine-tracker` Paket bereitgestellt, um die Pine Installation zu vereinfachen.) Die Anweisungen beschrieben in 'Portierung eines Pakets auf die `stable` Distribution' auf Seite 80 und 'Paketerzeugung' auf Seite 193 bieten einige Möglichkeiten, um ein Paket manuell zu paketieren.

Der Quellcode kann, muss aber nicht für Pakete in `contrib` und `non-free` Verzeichnissen, welche formal nicht Teil des Debian Systems sind, verfügbar sein.

2.2 Das Debian Paketverwaltungssystem

2.2.1 Überblick über Debian Pakete

Pakete enthalten im allgemeinen all die Dateien, welche nötig sind um eine Menge von zusammengehörigen Kommandos oder Eigenschaften zu implementieren. Es gibt zwei Typen von Debian Paketen:

- **Binärpakete**, welche ausführbare Programme enthalten, Konfigurationsdateien, man/info Seiten, Copyright Informationen und andere Dokumentationen. Diese Pakete werden in einem Debian spezifischen Archivformat verteilt (vergleiche 'Debian Paketformat' auf dieser Seite); sie zeichnen sich i.a. durch die `.deb` Dateierweiterung aus. Binärpakete können mit Debians `dpkg` Programm ausgepackt werden; Details sind in der Manual-Page beschrieben.

- **Quellpakete**, welche eine `.dsc` Datei enthalten, die das Quellpaket beschreibt (inklusive der Namen der folgenden Dateien), ebenso wie eine `.orig.tar.gz` Datei, welche den ursprünglichen unveränderten Quellcode in gzip-komprimiertem tar Format enthält und gewöhnlich eine `.diff.gz` Datei, die Debian spezifische Änderungen zu den Originalquellen enthält. Das Programm `dpkg-source` packt und entpackt Debian Quellpakete; Details sind in der Manual-Page enthalten.

Die Installation der Software durch das Paketsystem nutzt "Abhängigkeiten", welche sorgfältig vom Paketbetreuer bestimmt wurden. Diese Abhängigkeiten sind in der `control` Datei, die jedem Paket zugeordnet ist, enthalten. Das Paket, welches den GNU C Compiler enthält (`gcc`), "hängt" z.B. von dem Paketen `binutils` "ab", das den Linker und Assembler enthält. Versucht ein Benutzer `gcc` zu installieren, ohne zuvor `binutils` installiert zu haben, so wird das Paketverwaltungssystem (dpkg) die Fehlermeldung ausgeben, dass es `binutils` benötigt und die Installation von `gcc` abbrechen. (Dennoch kann dieses Verhalten vom Nutzer geändert werden; vergleiche `dpkg(8)`.) Für weitere Einzelheiten wird auf 'Paketabhängigkeiten' auf Seite 16 verwiesen.

Debian's Paketverwaltungstools können benutzt werden um:

- Pakete oder Teile von Paketen zu manipulieren und handzuhaben,
- dem Nutzer im Aufteilen von Paketen zu helfen, welche mittels kleiner Medien wie Disketten übertragen werden müssen,
- Entwickler beim Erzeugen von Paketarchiven zu helfen und um
- den Nutzern bei der Installation von Paketen, die sich auf entfernten Debian Archiven befinden, zu helfen.

2.2.2 Debian Paketformat

Ein Debian "Paket" oder eine Debian Archivdatei enthält ausführbare Dateien, Bibliotheken und Dokumentationen, welche einem bestimmten Programm oder einer Menge von zugehörigen Programmen zugeordnet sind. Normalerweise hat eine Debian Archivdatei einen Dateinamen der mit `.deb` endet. [1]

[1] Das `debian-installer` Projekt führte Dateinamen ein, die auf `.udeb` enden. Kurzum, es ist ein Mikro-`.deb` Format, das den Debian Richtlinien für freie Software nicht exakt folgen muss. So fehlt z.B. die Dokumentation und es wird nur vom `debian-installer` genutzt, dem neuen Debian Installationsprogramm, das für die Sarge Ausgabe entwickelt wurde. Die Dateiformate von `.deb` und `.udeb` sind identisch. Das Programm `udpkg`,

Die internen Einzelheiten dieses Debian Binärpaketformats werden in der `deb(5)` Manual-Page beschrieben. Da dieses interne Format in Zukunft geändert werden kann (zwischen verschiedenen Ausgaben von Debian), sollte stets `dpkg-deb(8)` für Änderungen der `.deb` Dateien verwendet werden.

Zumindest in der Sarge Distribution kann auf alle Debian Archivdateien mit den Standard Unix Kommandos `ar` und `tar` zugegriffen werden, auch wenn das `dpkg` Kommando nicht verfügbar ist.

2.2.3 Namenskonventionen für Debians Paketdateinamen

Die Debian Paketdateinamen folgen den Konventionen:

foo_Versionsnummer-DebianRevisionsnummer.deb

wobei *foo* für den Paketnamen steht. Zur Kontrolle kann man den Paketnamen, welcher einer bestimmten Debian Archivdatei (`.deb` Datei) zugeordnet ist, durch eine der folgenden Möglichkeiten bestimmen:

- betrachten der "Packages" Datei in dem Verzeichnis, wo sich die `.deb` Datei im Debian Archiv befand. Diese Datei enthält für jedes Paket einen Eintrag, welcher es genau beschreibt; das erste Feld in jedem Eintrag ist der formale Paketname.
- man benutzt das Kommando `dpkg --info` *foo_VVV-RRR*`.deb` (wobei *VVV* und *RRR* die Versions- bzw. Revisionsnummer des abgefragten Pakets sind). Dies zeigt unter anderen den Paketnamen an, welcher der zu entpackenden Archivdatei entspricht.

Der *VVV* Teil ist die Versionsnummer, die vom Entwickler des Programms vergeben wurde. Es gibt dafür kein allgemein gültiges Format, sowohl "19990513" als auch "1.3.8pre1" ist denkbar.

Der *RRR* Teil ist die Debian Revisionsnummer und wird vom Debian Entwickler angegeben (oder von einem Nutzer, wenn dieser das Paket selbst baut). Diese Nummer entspricht dem Revisionslevel des Debian Pakets; ein neues Revisionslevel kennzeichnet in der Regel Änderungen in Debians Makefile (`debian/rules`), in Debians Kontrolldatei (`debian/control`), in Installations- oder Deinstallationsskripten (`debian/p*`) oder in den Konfigurationsdateien, die mit dem Paket genutzt werden.

---

das mit `.udeb` Paketen umgehen kann, hat eingeschränktere Fähigkeiten als das Standard-`dpkg` und unterstützt weniger Paketbeziehungen. Der Unterschied im Namen besteht, weil die Debian Archivbetreuer nicht glücklich mit `.deb`s im Archiv sind, die den Richtlinien nicht folgen. Deshalb wurde ein unterschiedlicher Name gewählt um dies zu betonen und um es unwahrscheinlicher zu machen, dass Benutzer diese unabsichtlich auf einem richtigen System installieren. `.udeb`s werden nur in einer Initial Ramdisk während der Basisinstallation genutzt, um ein sehr beschränktes Debian System zu erstellen.

2.2.4 Bewahren der lokalen Konfiguration

Die durch den Nutzer konfigurierbaren Dateien werden durch Debian's "conffiles" Mechanismus bewahrt. Konfigurationsdateien (diese befinden sich im Allgemeinen in `/etc/`) werden in den `conffiles` innerhalb von Debian's Paketsystem angegeben. Das Paketverwaltungssystem garantiert, dass diese Dateien bei einer Paketaktualisierung (einem Upgrade) nicht überschrieben werden.

Wenn es möglich ist, das System ohne Modifizierungen von Dateien anzupassen, welche zu verschiedenen Debian Paketen gehören, so ist es in der Regel eine gute Idee, diese nicht zu verändern, selbst wenn es sich um "conffiles" handelt. Dies ermöglicht schnellere und sauberere Aktualisierungen.

Um zu bestimmen, welche Dateien während eines Upgrades bewahrt werden, kann man

```
dpkg --status Paket
```

ausführen und nach "Conffiles:" schauen.

Einzelheiten zum Inhalt einer Debian `conffiles` Datei sind im Debian Policy Manual, Abschnitt 11.7 (vergleiche 'Referenzen' auf Seite 199) zu finden.

2.2.5 Debian Wartungsskripte

Debian Wartungsskripte sind ausführbare Skripte, welche automatisch gestartet werden bevor oder nachdem ein Paket installiert wird. Zusammen mit einer Datei namens `control` sind all diese Dateien Teil des "control" Abschnitts einer Debian Archivdatei.

Die einzelnen Dateien sind:

preinst Dieses Skript wird ausgeführt, bevor das Paket aus der Debian Archivdatei (`.deb`) ausgepackt wird. Viele "preinst" Skripte beenden Dienste von Paketen, welche aktualisiert werden, bis deren Installation oder Upgrade vollzogen ist (d.h. nach der erfolgreichen Ausführung des "postinst" Skriptes).

postinst Dieses Skript schließt typischerweise jede nötige Konfiguration eines Paketes ab, nachdem es aus der Debian Archivdatei (`.deb`) ausgepackt wurde. Oft fragen "postinst" Skripte die Nutzer nach Daten, und/oder weisen sie darauf hin, dass, wenn sie die Standardwerte akzeptieren, sie später die Möglichkeit haben zurückzugehen und das Paket zu rekonfigurieren, sollte dies nötig sein. Viele "postinst" Skripte führen Kommandos aus, welche nötig sind, um Dienste zu starten oder nach der Installation oder dem Upgrade neu zu starten.

prerm Dieses Skript beendet typischerweise Daemonen, welche dem Paket zugeordnet sind. Es wird ausgeführt, bevor die zum Paket gehörenden Dateien gelöscht werden.

postrm Dieses Skript modifiziert typischerweise Links oder andere Dateien, welche dem Paket zugeordnet sind und/oder entfernen Dateien, welche von ihm erzeugt wurden. (Siehe auch 'Virtuelle Pakete' auf der nächsten Seite.)

Zur Zeit können all diese Kontrolldateien im Verzeichnis `/var/lib/dpkg/info` gefunden werden. Die auf das Paket `foo` bezogenen Dateien beginnen mit "foo" und haben die entsprechende Dateierweiterungen "preinst", "postinst", u.s.w. Die Datei `foo.list` in diesem Verzeichnis enthält alle Dateien, welche mit dem Paket `foo` installiert wurden. (Es ist zu beachten, dass die Position dieser Dateien eine interne `dpkg` Eigenschaft ist und sich in der Zukunft ändern kann.)

2.2.6 Paket Prioritäten

Jedem Debian Paket ist eine **Priorität** vom Distributionsbetreuer zugeordnet worden, um die Arbeit des Paketverwaltungssystems zu vereinfachen. Die Prioritäten sind:

- **Erforderliche** Pakete werden benötigt für die zuverlässige Funktionalität des Systems.

  Dies schließt alle Tools, welche nötig sind, um das System zu reparieren, ein. Diese Pakete dürfen nicht entfernt werden, andernfalls kann das System komplett versagen und man ist nicht einmal in der Lage `dpkg` zum Wiederherstellen zu nutzen. Systeme die nur die erforderlichen Pakete enthalten, sind wahrscheinlich ungeeignet für die meisten Aufgaben, jedoch kann der Systemadministrator jederzeit neue Software installieren.

- **Wichtige** Pakete sollten auf jedem Unix-artigen System gefunden werden.

  Andere Pakete ohne die das System nicht gut oder brauchbar arbeitet, haben diese Priorität. Dies schließt **nicht** Emacs, X11, TeX oder andere große Anwendungen ein. Diese Pakete erzeugen nur die nötige Infrastruktur.

- **Standard**pakete sind auf jedem Linuxsystem üblich, inklusive einem kleinen aber nicht zu sehr beschränkten textbasierten System.

  Dies ist der Standardinstallationsumfang, solange der Nutzer nichts anderes wählt. "Standard" enthält nicht viele große Anwendungen, aber es enthält Emacs (das ist mehr eine Infrastruktur als eine Anwendung) und eine geeignete Teilmenge von TeX und LaTeX (sofern dies ohne X möglich ist).

- **Optionale** Pakete enthalten all diese, welche man vernünftigerweise installieren möchte, auch wenn man damit nicht vertraut ist und keine speziellen Anforderungen daran hat.

  Dies schließt X11, eine vollständige TeX Distribution und viele Anwendungen mit ein.

- **Zusätzliche** Pakete sind entweder nicht mit anderen Paketen mit höherer Priorität verträglich, sind wahrscheinlich nur nützlich, wenn man sie bereits näher kennt oder haben spezielle Anforderungen, die sie für "Optional" ungeeignet machen.

Beachten Sie die Unterschiede zwischen “Priority: required”, “Section: base” und “Essential: yes” in der Paketbeschreibung. “Section: base” bedeutet, dass dieses Paket vor allen anderen in einem neuen System installiert wird. Die meisten der Pakete mit “Section: base” enthalten “Priority: required” oder zumindest “Priority: important” und viele von diesen sind mit “Essential: yes” versehen. “Essential: yes” bedeutet, dass das Paket die Angabe einer bestimmten Option an das Paketmanagementsystem wie `dpkg` erfordert, wenn es aus dem System entfernt werden soll. Beispielsweise sind `libc6`, `mawk` und `makedev` Pakete mit “Priority: required” und “Section: base” aber enthalten nicht “Essential: yes”.

2.2.7 Virtuelle Pakete

Ein virtuelles Paket ist ein allgemeiner Name, welcher für eine Gruppe von Paketen steht, die alle ähnliche Funktionalitäten bereitstellen. Die Programme `tin` und `trn` sind beispielsweise beide News-Reader und einer von beiden sollte deshalb die Abhängigkeiten eines Programms, welches einen News-Reader im System benötigt um richtig funktionieren zu können, erfüllen. Deshalb sagt man, dass beide das “virtuelle Paket” `news-reader` bereitstellen.

Analog stellen `exim` und `sendmail` die Funktionalität eines Mail Transport Agent bereit. Deshalb wird das virtuelle Paket `mail-transport-agent` von beiden angeboten. Wenn eines dieser Programme installiert ist, dann wird jedes Programm, das von der Installation eines Mail-Transport-Agent abhängt, mit der Existenz dieses virtuellen Paketes zufrieden sein.

Debian besitzt einen Mechanismus so dass, wenn mehr als ein Paket, welches das selbe virtuelle Paket bereitstellt, auf dem System installiert ist, der Systemadministrator ein Programm bevorzugt auswählen kann. Das entsprechende Kommando ist `update-alternatives` und wird genauer in ‘Alternative Befehle’ auf Seite 83 beschrieben.

2.2.8 Paketabhängigkeiten

Das Debian Paketsystem enthält “Paketabhängigkeiten”, welche (in einem einzelnen Eintrag) festhalten, wie ein Programm A unabhängig von der Existenz eines Programms B auf einem System arbeiten kann.

- Paket A **hängt** von Paket B **ab**, wenn B unbedingt installiert sein muss, um A starten zu können. In einigen Fällen hängt A nicht nur von B, sondern einer speziellen Version von B ab. In diesem Fall ist die Versionsabhängigkeit im Allgemeinen eine untere Schranke, d.h. A hängt von einer beliebigen Version von B ab, welche aktueller als eine angegebene Version ist.

- Paket A **empfiehlt** Paket B, wenn der Paketbetreuer meint, dass die meisten Nutzer A nicht ohne die von B bereitgestellte Funktionalität haben wollen.

- Paket A **schlägt** Paket B **vor**, wenn B Dateien enthält, welche sich auf die Funktionalität von A beziehen (und diese eventuell erweitern).

- Paket A **kollidiert** mit Paket B, wenn A nicht funktioniert, sofern B auf dem System installiert ist. Sehr oft bestehen solche **Konflikte** darin, dass A Dateien enthält, die gegenüber denen in B verbessert wurden. Der "Konflikt" Status wird oft mit "ersetzt" kombiniert.
- Paket A **ersetzt** Paket B, wenn Dateien, die von B installiert wurden, von A entfernt und (in einigen Fällen) durch Dateien in A überschrieben werden.
- Paket A **unterstützt** Paket B, wenn alle Dateien und Funktionalitäten von B in A verfügbar sind. Dieser Mechanismus steht für Nutzer mit geringem Plattenplatz zur Verfügung, um nur den Teil von A zu installieren, welcher absolut nötig ist.

Weitere detaillierte Informationen zur Verwendung dieser Terme können im *Packaging Manual* und dem *Policy Manual* gefunden werden.

Es ist zu beachten, dass `dselect` eine bessere Kontrolle über Pakete, die mit **empfiehlt** und **schlägt vor** spezifiziert werden, bietet als `apt-get`, was einfach alle **hängt ab** Pakete wählt und **empfiehlt** und **schlägt vor** ignoriert. Beide Programme nutzen in aktuellen Versionen APT als Back-end.

2.2.9 Die Bedeutung von "pre-depends"

"Pre-depends" ist eine spezielle Abhängigkeit. Im Falle eines gewöhnlichen Pakets entpackt `dpkg` die Archivdatei (d.h. dessen `.deb` Datei) unabhängig davon, ob die Dateien, von denen es abhängt, im System existieren oder nicht. Entpacken bedeutet im wesentlichen, dass `dpkg` die Dateien aus der Archivdatei extrahiert und korrekt im Dateisystem platziert. **Hängen** diese Pakete von der Existenz anderer Pakete auf dem System **ab**, dann wird sich `dpkg` weigern, die Installation zu beenden (durch Ausführen des "konfigurieren" Schritts), bis die anderen Pakete installiert sind.

Dennoch gibt es einige Pakete, bei denen sich `dpkg` sogar weigert, sie zu entpacken, bis gewisse Abhängigkeiten aufgelöst sind. Solche Pakete hängen vom Vorhandensein anderer Pakete im Sinne von "pre-depend" ab. Das Debian Projekt führte diesen Mechanismus ein, um das System sicher vom `a.out` auf das ELF Format zu aktualisieren, wobei die **Reihenfolge** in welcher die Pakete entpackt werden, bedeutend ist. Es gibt weitere große Update Situationen, wo diese Methode nützlich ist, z.B. für Pakete mit der "erforderlich" Priorität und deren libc Abhängigkeit.

Erneut wird für detailliertere Informationen über dies auf das *Packaging Manual* verwiesen.

2.2.10 Paket Status

Der Paket Status kann "unbekannt", "installieren", "entfernen", "säubern" oder "halten" sein. Diese "gewünschten" Werte kennzeichnen, was der Nutzer mit einem Paket beabsichtigte (entweder durch Anwahl des "[A]uswählen" Punktes in `dselect` oder durch direkten Aufruf von `dpkg`).

Deren Bedeutung ist:

- **unbekannt** - der Nutzer hat niemals angegeben, ob er das Paket will.
- **installieren** - der Nutzer will das Paket installiert oder aktualisiert haben.
- **entfernen** - der Nutzer will das Paket entfernen lassen, ohne das existierende Konfigurationsdateien gelöscht werden.
- **säubern** - der Nutzer will das Paket komplett entfernt haben, inklusive der Konfigurationsdateien.
- **halten** - der Nutzer will das Paket nicht verarbeiten lassen, d.h. er möchte die aktuelle Version im aktuellen Status belassen, unabhängig vom Wert.

2.2.11 Zurückhalten von Paketen von einem Upgrade

Es gibt zwei Mechanismen zum Zurückhalten von Paketen von einem Upgrade, durch `dpkg` oder beginnend mit Woody durch APT.

Mit `dpkg` ist zuerst die Paketauswahlliste zu exportieren:

```
dpkg --get-selections \* > Paketauswahl.txt
```

Dann muss die erzeugte Datei *`Paketauswahl.txt`* editiert werden, indem die Zeile, welche das zu haltende Paket, z.B. `libc6`, enthält, von:

```
libc6                           install
```

auf:

```
libc6                           hold
```

geändert wird. Nach dem Speichern ist die Datei in die `dpkg` Datenbank zurückzuladen mit:

```
dpkg --set-selections < Paketauswahl.txt
```

Kennt man den Paketnamen des zu haltenden Pakets, kann man auch einfach

```
echo libc6 hold | dpkg --set-selections
```

starten. Wann immer der Installations Prozess dieses Paket bearbeitet (zu upgraden versucht), hält er es zurück.

Der selbe Effekt kann mit `dselect` erreicht werden. Dazu ist einfach der Punkt [A]uswählen und dann das Paket zu wählen, dessen Status beibehalten werden soll sowie schließlich '=' (oder 'H') zu drücken. Die Änderungen werden sofort aktiv, nachdem das [A]uswählen Menü beendet wird.

Das APT System in der Woody Distribution hat einen neuen "Alternativen" Mechanismus zum Halten von Paketen während des Archivabfrageprozesses unter Verwendung von `Pin-Priority`. Vergleiche die Manual-Page `apt_preferences(5)` sowie `http://www.debian.org/doc/manuals/apt-howto/` oder das `apt-howto` Paket; 'Überblick über `/etc/apt/preferences`' auf Seite 71 enthält auch eine kurze Erläuterung.

2.2.12 Quellpakete

Quellpakete befinden sich in einem Verzeichnis namens `source` und können entweder manuell heruntergeladen werden oder durch

```
apt-get source foo
```

(vergleiche die `apt-get(8)` Manual-Page wie man APT dazu bringt, dies zu tun).

2.2.13 Erzeugen von Binär- aus Quellcodepaketen

Für ein Paket *foo* benötigt man alle *foo*`_*.dsc`, *foo*`_*.tar.gz` und *foo*`_*.diff.gz` Dateien, um die Quellen zu übersetzen (Bemerkung: es gibt keine `.diff.gz` Datei für ein natives Debian Paket).

Sind diese Dateien vorhanden und ist das `dpkg-dev` Paket installiert, so extrahiert das Kommando

```
$ dpkg-source -x foo_version-revision.dsc
```

das Paket in ein Verzeichnis namens *foo-version*.

Zum Erzeugen des Binärpakets ist folgendes auszuführen:

```
$ cd foo-version
$ su -c "apt-get update; apt-get install fakeroot"
$ dpkg-buildpackage -rfakeroot -us -uc
```

Und danach

```
# su -c "dpkg -i ../foo_version-revision_arch.deb"
```

zum Installieren des neu gebildeten Pakets. Vergleiche ‘Portierung eines Pakets auf die `stable` Distribution’ auf Seite 80.

2.2.14 Erzeugen neuer Debian Pakete

Für detaillierte Informationen zum Erzeugen neuer Pakete sollte der *New Maintainers’ Guide* gelesen werden, verfügbar im `maint-guide` Paket, oder unter `http://www.debian.org/doc/manuals/maint-guide/`.

2.3 Aktualisierung eines Debian Systems

Einer von Debian's Vorzügen ist die Unterstützung eines konsistenten Upgrade Wegs sowie ein sicherer Upgrade Prozess und es wird stets versucht, eine ältere Ausgabe problemlos zu aktualisieren. Pakete warnen den Nutzer, sollten bedeutende Bemerkungen während des Upgrade Prozesses auftreten und bieten oft eine Lösung zu einem möglichen Problem.

Man sollte auch die Release Notes lesen, das Dokument, das die Details zu spezifischen Upgrades enthält. Es wird mit allen Debian CDs ausgeliefert und ist im WWW unter `http://www.debian.org/releases/stable/releasenotes` oder `http://www.debian.org/releases/testing/releasenotes` verfügbar.

Eine praktische Anleitung zum Aktualisieren wird in 'Debian Paketverwaltung' auf Seite 65 bereitgestellt. Dieser Abschnitt beschreibt die grundlegenden Einzelheiten.

2.3.1 Methoden zum Aktualisieren eines Debian Systems

Man kann immer einfach einen anonymen FTP oder `wget` Aufruf zu einem Debian Archiv starten, die Verzeichnisse durchsehen, bis man die gewünschte Datei gefunden hat, diese herunterladen und schließlich mittels `dpkg` installieren. (Es ist zu beachten, dass `dpkg` die aktualisierten Dateien immer in das korrekte Verzeichnis installiert, auch in einem laufenden System.) Manchmal kommt es dennoch vor, dass ein überarbeitetes Paket die Installation einer neuen Version eines anderen Paketes erfordert. In diesem Fall schlägt die Installation fehl, wenn das andere Programm nicht installiert ist.

Viele Personen finden dieses manuelle Vorgehen zu Zeit aufwendig, da Debian sich so schnell entwickelt – typischerweise werden ein Dutzend oder mehr Pakete jede Woche hochgeladen. Diese Zahl ist kurz vor einer neuen Veröffentlichung noch größer. Um dies besser handhaben zu können, bevorzugen viele Personen ein automatisches Programm zum Upgraden. Einige spezialisierte Paketmanagement Tools sind für diesen Zweck verwendbar.

2.3.2 Überblick über Paketverwaltungstools

Das Debian Paketverwaltungssystem hat zwei Aufgaben: die Manipulation der Paketdatei und das Herunterladen von Paketdateien aus dem Debian Archiv. `dpkg` erfüllt die erste Aufgabe, APT und `dselect` die letztere.

2.3.3 `dpkg`

Dies ist das wichtigste Programm zum Manipulieren von Paketdateien. Für eine ausführliche Beschreibung ist `dpkg(8)` zu lesen.

`dpkg` wird mit einigen wichtigen Programmen ausgeliefert.

- `dpkg-deb`: Manipuliert `.deb` Dateien. `dpkg-deb(1)`

- `dpkg-ftp`: Ein älteres Programm zum Herunterladen von Paketen. `dpkg-ftp(1)`
- `dpkg-mountable`: Ein älteres Programm zum Herunterladen von Paketen. `dpkg-mountable(1)`
- `dpkg-split`: Teilt ein größeres Paket in kleinere Dateien auf. `dpkg-split(1)`

`dpkg-ftp` und `dpkg-mountable` wurden durch das APT System ersetzt.

2.3.4 APT

APT, das zukunftsweisende Paket-Tool (Advanced Packaging Tool) ist eine fortschrittliche Schnittstelle zu Debian's Paketsystem und besteht aus verschiedenen Programmen, deren Namen oft mit "apt-" beginnen. `apt-get`, `apt-cache` und `apt-cdrom` sind die Kommandozeilen Tools zum Umgang mit Paketen. Diese werden auch von anderen Programmen, wie `dselect` und `aptitude` genutzt.

Für weitere Informationen ist das `apt` Paket zu installieren und `apt-get(8)`, `apt-cache(8)`, `apt-cdrom(8)`, `apt.conf(5)`, `sources.list(5)`, `apt_preferences(5)` (Woody) sowie `/usr/share/doc/apt/guide.html/index.html` zu lesen.

Weitere Informationen finden sich im APT HOWTO (http://www.debian.org/doc/manuals/apt-howto/). Dazu kann in Sarge `apt-howto-de` installiert werden und steht dann unter `/usr/share/doc/Debian/apt-howto/` zur Verfügung (Woody enthält nur die englische Version `apt-howto`).

`apt-get upgrade` und `apt-get dist-upgrade` installieren nur die Pakete, die unter "Depends:" ("hängt ab:") aufgeführt sind und übersieht alle unter "Recommends:" ("empfiehlt:") und "Suggests:" ("schlägt vor:") gelisteten Pakete. Um dies zu vermeiden, sollte `dselect` verwendet werden.

2.3.5 `dselect`

Dieses Programm besitzt eine Menüoberfläche zu Debian's Paketverwaltungssystem. Es ist besonders für die Erstinstallation und große Upgrades nützlich. Vergleiche '`dselect`' auf Seite 67.

Für weitere Informationen sollte das `install-doc` Paket installiert und `/usr/share/doc/install-doc/dselect-beginner.en.html` oder dselect Documentation for Beginners (http://www.debian.org/releases/woody/i386/dselect-beginner) gelesen werden.

2.3.6 Aktualisieren eines laufenden Systems

Der Kernel (das Dateisystem) in Debian Systemen unterstützt das Ersetzen von Dateien auch während sie benutzt werden.

Es wird auch ein Programm namens `start-stop-daemon` bereitgestellt, das verwendet wird, um Daemonen während des Bootens zu starten oder zum Stoppen von Daemonen, wenn das Kernel-Runlevel geändert wird (z.B. von Mehrbenutzer- zu Einzelbenutzermodus oder zu "halt"). Das gleiche Programm wird von Installationsskripten genutzt, wenn ein Paket, das einen Daemon enthält, installiert wird, um laufende Daemonen zu stoppen und bei Bedarf neuzustarten.

Es wird darauf hingewiesen, dass das Debian System den Einzelnutzermodus nicht erfordert, um ein laufendes System zu aktualisieren.

2.3.7 Heruntergeladene und zwischengespeicherte `.deb` Archiv Dateien

Hat man Paketdateien manuell auf die Festplatte heruntergeladen (was nicht unbedingt notwendig ist, vergleiche die obige Beschreibung von `dpkg-ftp` oder APT), dann kann man die `.deb` Dateien nach der Installation der Pakete aus dem System entfernen.

Wenn APT verwendet wird, so werden diese Dateien im `/var/cache/apt/archives` Verzeichnis zwischengespeichert. Sie können nach der Installation gelöscht werden (`apt-get clean`) oder auf einen anderen Rechner ins `/var/cache/apt/archives` Verzeichnis kopiert werden, um das Herunterladen während mehrerer Installationen zu vermeiden.

2.3.8 Aufbewahren des Datensatzes für Upgrades

`dpkg` bewahrt einen Datensatz aller Pakete, die ausgepackt, konfiguriert, entfernt und/oder gesäubert wurden, auf. Jedoch wird (zur Zeit) kein Protokoll der Terminaleingaben, während der Paketmanipulation, erzeugt.

Der einfachste Weg, dieses Problem zu umgehen, ist Sitzungen von `dpkg`, `dselect`, `apt-get`, etc. innerhalb des `script(1)` Programms ablaufen zu lassen.

2.4 Der Debian Bootvorgang

2.4.1 Das `init` Programm

Wie alle Unices, wird Debian durch das Programm `init` gestartet. Die Konfigurationsdatei für `init` (dies ist `/etc/inittab`) gibt an, dass das erste zu startende Skript `/etc/init.d/rcS` ist. Dieses Skript startet alle anderen Skripte in `/etc/rcS.d/`, entweder indem diese eingebunden oder explizit als Unterprozess aufgerufen werden, je nach Dateierweiterung. Diese Skripte initialisieren das System indem sie z.B. Dateisysteme überprüfen und einbinden, Module laden, Netzwerk Dienste starten, die Uhrzeit setzen, u.a. Danach werden zwecks Kompatibilität die Dateien (mit Ausnahme der mit einem '.' im Dateinamen) in `/etc/rc.boot/` ausgeführt. Jedes Skript in diesem Verzeichnis ist normalerweise dem Systemadministrator vorbehalten, die Verwendung dieser in Paketen wird missbilligt. Vergleichen Sie 'Hinweise zur System-Inititalisierung' auf Seite 117 und System run

levels and init.d scripts (http://www.debian.org/doc/debian-policy/ch-opersys.html#s-sysvinit) in den Debian Richtlinien für weitere Informationen.

2.4.2 Runlevel

Nach dem Bootprozess führt `init` alle Startskripte in einem durch das Standard-Runlevel festgelegten Verzeichnis aus. (Dieses Runlevel wird durch den Eintrag `id` in `/etc/inittab` festgelegt). Wie viele System V kompatible Unices hat Linux 7 Runlevel:

- 0 (Anhalten des Systems),
- 1 (Einzelnutzer Modus),
- 2 bis 5 (verschiedene Mehrbenutzer Modi) und
- 6 (Neustart des Systems).

Für Debian Systeme gilt `id=2`, was bedeutet, dass das Standard-Runlevel 2 sein wird, wenn der Mehrbenutzer Modus aktiv ist und die Skripte in `/etc/rc2.d/` werden ausgeführt.

In Wirklichkeit sind die Skripte in den Verzeichnissen `/etc/rcN.d/` nur symbolische Links zu Skripten in `/etc/init.d/`. Dennoch werden die **Namen** der Dateien in jedem der `/etc/rcN.d/` Verzeichnisse individuell gewählt, um anzugeben **wie** die Skripte in `/etc/init.d/` gestartet werden. Speziell werden bevor ein Runlevel aktiv wird, alle Skripte die mit 'K' beginnen ausgeführt; diese Skripte beenden Dienste. Danach werden alle Skripte die mit 'S' beginnen gestartet; diese Skripte starten Dienste. Die zweistellige dem 'K' oder 'S' folgende Nummer bestimmt die Reihenfolge der Ausführung. Skripte mit kleinerer Nummer werden zuerst ausgeführt.

Dieses Vorgehen funktioniert, da die Skripte in `/etc/init.d/` alle ein Argument akzeptieren, das entweder "start", "stop", "reload", "restart" oder "force-reload" sein kann und eine dem Argument entsprechende Aktion ausführen (starten, stoppen, neuladen, neustarten, erzwinge Neuladen). Diese Skripte können auch ausgeführt werden, nachdem das System gebootet wurde, um verschiedene Prozesse zu kontrollieren.

Zum Beispiel führt das Argument "reload" im Kommando

```
# /etc/init.d/sendmail reload
```

dazu, dass der sendmail Daemon ein Signal zum erneuten Einlesen der Konfigurationsdatei erhält.

2.4.3 Anpassen des Bootvorgangs

Debian verwendet kein BSD typisches `rc.local` Verzeichnis, um den Bootvorgang anzupassen; stattdessen wird folgender Mechanismus angeboten.

Angenommen `foo` sei ein Skript, das während des Startvorgangs oder beim Übergang in ein bestimmtes (System V) Runlevel aufgerufen werden soll. Dann sollte der Systemadministrator:

1 Das Skript `foo` in das Verzeichnis `/etc/init.d/` verschieben.

2 Das Debian Kommando `update-rc.d` mit entsprechenden Argumenten starten, um Links zwischen den (Kommandozeilen spezifischen) Verzeichnissen `rc?.d` und `/etc/init.d/foo` anzulegen. Dabei bezeichnet *?* eine Nummer von 0 bis 6, die einem der System V Runlevel entspricht.

3 Das System neu booten.

Das Kommando `update-rc.d` setzt Links zwischen Dateien im Verzeichnis `rc?.d` und dem Skript in `/etc/init.d/`. Jeder Link beginnt mit einem 'S' oder 'K', gefolgt von einer Nummer, gefolgt vom Namen des Skripts. Beim Wechsel in das Runlevel *N*, werden Skripte in `/etc/rcN.d/` die mit 'K' beginnen mit `stop` als Argument ausgeführt, gefolgt von den mit 'S' beginnenden Skripten in `/etc/rcN.d/` mit `start` als Argument.

Man kann z.B. das Skript `foo` beim Booten ausführen lassen, indem man es nach `/etc/init.d/` verschiebt und die Links mit `update-rc.d foo defaults 19` erstellt. Das Argument `defaults` bezieht sich auf das Standard-Runlevel, welches zwischen 2 und 5 liegt. Das Argument `19` sichert, dass `foo` vor allen Skripten, welche die Nummern 20 oder größer enthalten, gestartet wird.

2.5 Unterstützung von Vielfalten

Debian unterstützt verschiedene Möglichkeiten zum Anpassen des Systems, ohne das System zu beeinträchtigen.

- `dpkg-divert`, vergleiche 'Der `dpkg-divert` Befehl' auf Seite 83.
- `equivs`, vergleiche 'Das `equivs` Paket' auf Seite 83.
- `update-alternative`, vergleiche 'Alternative Befehle' auf Seite 83.
- `make-kpkg` kann viele Boot-Loader anpassen. Vergleiche `make-kpkg(1)` und 'Die Debian Standard Methode' auf Seite 88.

Alle Dateien unter `/usr/local/` gehören dem Systemadministrator und Debian wird sie nicht verändern. Viele (oder alle) Dateien unter `/etc` sind Konfigurationsdateien und Debian wird sie nicht während eines Upgrades überschreiben, es sei denn der Systemadministrator erlaubt dies ausdrücklich.

2.6 Internationalisierung

Das Debian System ist internationalisiert und bietet Unterstützung zur Ein- und Ausgabe von Zeichen in vielen Sprachen, beides in der Konsole und unter X. Viele Texte, Manual-Pages und

Systemausgaben wurden in eine ständig wachsende Anzahl von Sprachen übersetzt. Während der Installation fordert Debian den Nutzer zur Wahl der Installationssprache (und manchmal eines lokalen Dialekts) auf.

Sollte das installierte System nicht alle benötigten Eigenschaften der Sprache unterstützen, soll eine andere Sprache gewählt werden oder wurde eine neue Tastatur angeschlossen, um die Sprache zu unterstützen, vergleiche 'Lokalisation und Sprachen' auf Seite 147.

2.7 Debian und der Kernel

Vergleiche 'Der Linux Kernel unter Debian' auf Seite 87.

2.7.1 Kompilierung eines Kernel aus Debian fremden Quellen

Man sollte die Debian Politik bezüglich der Header verstanden haben, bevor man startet.

Die Debian C Bibliotheken wurden mit der aktuellsten **stabilen** Version der **Kernelheader** erstellt.

Zum Beispiel nutzte die Debian 1.2 Ausgabe Version 5.4.13 der Header. Dieses Vorgehen unterscheidet sich von den Linux Kernelquellpaketen, die auf allen Linux FTP Archiv Seiten verbreitet werden, welche aktuellere Versionen der Header verwenden. Die Kernelheader aus den Kernelquellen befinden sich in `/usr/include/linux/include/`.

Wenn es nötig ist, ein Programm mit aktuelleren Kernelheadern als in `libc6-dev` zu übersetzen, so muss `-I/usr/src/linux/include/` zur Kommandozeile beim Kompilieren hinzugefügt werden. Dies ist z.B. für das Paketieren des automounter Daemon (`amd`) von Bedeutung. Als neue Kernel einige NFS bezogene Internals änderten, musste dies `amd` mitgeteilt werden. Dies erforderte die Einbindung der aktuellsten Kernelheader.

2.7.2 Tools zum Erzeugen angepasster Kernel

Nutzern die einen angepassten Kernel erzeugen wollen (oder müssen), wird empfohlen, das Paket `kernel-package` herunterzuladen. Dieses Paket enthält das Skript zur Kernelerstellung und bietet die Möglichkeit, ein Debian kernel-image Paket einfach durch Aufruf von

```
# make-kpkg kernel_image
```

im Kernelquellverzeichnis zu starten. Hilfe ist durch Ausführung von

```
# make-kpkg --help
```

verfügbar und durch die Manual-Page `make-kpkg(8)` sowie 'Der Linux Kernel unter Debian' auf Seite 87.

Nutzer müssen den Quellcode für den aktuellsten Kernel (oder den Kernel ihrer Wahl) separat vom bevorzugten Linux Archiv herunterladen, wenn kein `kernel-source-`*`version`* Paket (dabei steht *version* für die Kernel Version) vorhanden ist. Das Debian `initrd` Bootskript erfordert einen speziellen Kernel Patch namens `initrd`; vergleiche `http://bugs.debian.org/149236`.

Detaillierte Anweisungen zur Benutzung des `kernel-package` Pakets sind in der Datei `/usr/share/doc/kernel-package/README.gz` zu finden.

2.7.3 Alternative Boot Loader

Zur Verwendung von alternativen Boot Loader wie `grub` oder `loadlin` ist der kompilierte Linux Kernel `bzimage` in ein anderes Verzeichnis (wie `/boot/grub` oder auf eine MS-DOS Partition) zu kopieren.

2.7.4 Erzeugen von Boot Disketten

Die Erstellung einer angepassten Boot Diskette wird durch das Debian Paket `boot-floppies` unterstützt, das normalerweise im `admin` Abschnitt des Debian FTP Archivs gefunden wird. Shellskripte in diesem Paket erzeugen Boot Disketten im `syslinux` Format. Dies sind MS-DOS formatierte Disketten, deren Master-Bootrecord so modifiziert wurde, dass sie direkt Linux (oder irgend ein anderes Betriebssystem, das in der `syslinux.cfg` Datei auf der Diskette definiert wurde) booten können. Andere Skripte in diesem Paket erzeugen Notfall Disketten und können auch die Basisdisketten neu erzeugen.

Mehr Informationen darüber können in der `/usr/doc/boot-floppies/README` Datei nach der Installation des `boot-floppies` Paketes gefunden werden.

2.7.5 Spezielle Regeln für den Umgang mit Modulen

Debian's `modconf` Paket enthält ein Shellskript (`/usr/sbin/modconf`), dass zur Anpassung der Modulkonfiguration genutzt werden kann. Dieses Skript besitzt eine Menü basierte Schnittstelle, die den Nutzer nach Einzelheiten über ladbare Gerätetreiber im System fragt. Die Antworten werden benutzt, um die Datei `/etc/modules.conf` (welche Aliase enthält sowie andere Argumente, die in Verbindung mit verschiedenen Modulen verwendet werden müssen) anzupassen, auf Grund von Dateien in `/etc/modutils/` und `/etc/modules` (die die Module auflistet, die zur Bootzeit geladen werden müssen).

Wie die (neuen) `Configure.help` Dateien, die nun verfügbar sind, um angepasste Kernel zu unterstützen, kommt das `modconf` Paket mit einer Reihe von Hilfe Dateien (in `/usr/lib/modules_help/`), die detaillierte Informationen über passende Argumente für jedes Modul enthalten. Vergleiche 'Der modularisierte Kernel 2.4' auf Seite 89 für Beispiele.

2.7.6 Deinstallation eines alten Kernel Pakets

Das `kernel-image-`*`NNN`*`.prerm` Skript überprüft, ob der aktuell laufende Kernel mit dem zu entfernenden Kernel übereinstimmt. Deshalb können ungewünschte kernel-image Pakete sicher mittels

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

entfernt werden. (*NNN* ist durch die entsprechende Kernelversion und Revisionsnummer zu ersetzen.)

Kapitel 3

Debian System-Installationshinweise

Offizielle Dokumentation zur Installation von Debian ist unter `http://www.debian.org/releases/stable/` und `http://www.debian.org/releases/stable/installmanual` zu finden.

Die Entwicklerversionen davon sind `http://www.debian.org/releases/testing/` und `http://www.debian.org/releases/testing/installmanual` (befinden sich in Arbeit und sind manchmal eventuell nicht vorhanden).

Obwohl dieses Kapitel ursprünglich für die Potato Installation geschrieben wurde, ist der größte Teil des Inhalts auf das Woody Installationsprogramm aktualisiert worden; beide sind sehr ähnlich. Da Sarge ein komplett anderes Installationsprogramm verwenden wird, sollten Sie dies nur als Anhaltspunkt für die Sarge Installation verstehen.

3.1 Allgemeine Linux System-Installationshinweise

Vergessen Sie nicht unter `http://www.debian.org/CD/netinst/` zu schauen, wenn Sie nach einem kompakten CD Image des Debian Installers suchen.

Die Verwendung der `testing` oder `unstable` Ausgabe von Debian führt zum erhöhten Risiko, auf verschiedene Fehler zu stoßen. Dieses Risiko kann durch ein Multi-Boot Schema gemanagt werden, bei dem eine stabilere Ausgabe von Debian genutzt wird. Alternativ kann auch ein netter Trick mit `chroot` und der stabileren Version wie in '`chroot`' auf Seite 114 beschrieben, genutzt werden. Letzteres ermöglicht die Nutzung verschiedener Versionen von Debian simultan auf verschiedenen Konsolen.

3.1.1 Grundlegendes zur Hardware Kompatibilität

Linux ist zur meisten PC Hardware kompatibel und kann auf fast jedem Rechner installiert werden. Es war für mich so einfach wie eine Windows 95/98/Me Installation. Die Liste unterstützter Hardware wächst beständig.

Sollten Sie ein Laptop besitzen, schauen Sie unter Linux auf Laptops (http://www.linux-laptop.net/) für Hinweise zur Installation zu verschiedenen Marken und Modellen.

Meine Empfehlung zur Desktop PC Hardware ist "sei konservativ":

- SCSI statt IDE für die Arbeit, IDE/ATAPI HD für private Nutzung.
- IDE/ATAPI CD-ROM (oder CD-RW).
- PCI statt ISA, insbesondere für die Netzwerkkarte (NIC).
- Verwende eine billige NIC. Tulip für PCI und NE2000 für ISA sind geeignet.
- Vermeide PCMCIA Karten (Notebook) für die erste Linux Installation.
- Keine USB Tastatur, Maus, ... es sei denn Sie lieben die Herausforderung.

Sollten Sie einen langsamen Rechner haben, so ist das vorübergehende Entfernen und Einbauen der Festplatte in einen schnelleren Rechner für die Installation eine gute Idee.

3.1.2 Bestimmung der PC Hardware und des Chipsatzes

Während der Installation wird man nach der eingebauten Hardware oder dem Chipsatz des PCs gefragt. Manchmal sind diese Informationen nicht leicht zu finden. Eine Möglichkeit ist:

1. Öffnen des PC Gehäuses.
2. Bestimmen der Aufdrucke auf den großen Chips auf der Graphik- und Netzwerkkarte, sowie den Chips nahe der seriellen und IDE Anschlüsse.
3. Bestimmen der Kartennamen, die auf der Rückseite der PCI und ISA Karten aufgedruckt sind.

3.1.3 Bestimmung der PC Hardware mit Debian

Die folgenden Kommandos eines Linux Systems sollten Aufschluss über aktuelle Hardware und deren Konfiguration geben.

```
$ lspci -v | pager
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
$ pager /proc/bus/usb/devices
```

Diese Kommandos können während der Installation in der Konsole nach Drücken von Alt-F2 gestartet werden.

Für USB Geräte werden die Geräteklassen in `/proc/bus/usb/devices` mit `Cls=`*nn* aufgeführt:

- Cls=00 : nicht benutzt
- Cls=01 : Audio (Lautsprecher, etc.)
- Cls=02 : Kommunikation (Modem, Netzwerkkarte, ...)
- Cls=03 : HID (Human Interface Device (Schnittstellen): Tastatur, Maus, Joystick)

- Cls=07 : Drucker
- Cls=08 : Massenspeicher (Diskettenlaufwerk, CD/DVD Laufwerk, Festplatte, Flash Speicher, ...)
- Cls=09 : Hub (USB Hub)
- Cls=255 : Hersteller spezifisch

Ist die Geräteklasse eines Gerätes nicht 255, so wird es von Linux unterstützt.

3.1.4 Bestimmen der PC Hardware mit anderen Betriebssystemen

Hardware Informationen können auch von anderen Betriebssystemen erhalten werden:

Man kann eine andere kommerzielle Linux Distribution installieren. Die Hardwareerkennung ist bei diesen oft besser, als dies bei Debian zur Zeit der Fall ist. (Diese Situation sollte ausgewogener sein, wenn `debian-installer` mit Sarge eingeführt wird.)

In einem Windows System kann die Hardware Konfiguration durch Rechts-Klick auf "Mein Computer" unter Eigenschaften / Gerätemanager erhalten werden. Die Ressourcen wie IRQ, I/O Portadressen sowie DMA sollten festgehalten werden. Einige alte ISA Karten müssen eventuell unter DOS konfiguriert werden.

3.1.5 Ein Lilo Mythos

"Lilo ist auf 1024 Zylinder beschränkt." Falsch!

Das neuere `lilo` das nach Debian Potato verwendet wird, bietet Unterstützung für lba32. Ist das BIOS des Motherboards aktuell genug um lba32 zu unterstützen, so sollte `lilo` in der Lage sein, auch von außerhalb der alten 1024 Zylindergrenze zu booten.

Es muss nur sichergestellt werden, dass die Zeile "lba32" irgendwo am Anfang der `lilo.conf` Datei hinzugefügt wird, wenn noch eine alte `lilo.conf` vorhanden ist. Man vergleiche `/usr/share/doc/lilo/Manual.txt.gz`.

3.1.6 GRUB

Der neue Bootloader `grub` des GNU Hurd Projekts kann auf einem Debian Woody System wie folgt installiert werden:

```
# apt-get update
# apt-get install grub-doc
# mc /usr/share/doc/grub-doc/html/
... lesen Sie den Inhalt
# apt-get install grub
# pager /usr/share/doc/grub/README.Debian
... lesen Sie dies :)
```

Um das GRUB Menü anzupassen, ist `/boot/grub/menu.lst` zu editieren. Man vergleiche 'How do I set boot parameters (GRUB)' auf Seite 98 zum Setzen von Bootparametern während des Bootvorgangs, da sich dies leicht von der `lilo` Konfiguration unterscheidet.

3.1.7 Wahl der Boot Disketten

In Potato nutzte ich IDEPCI für die normale Desktopinstallation. In Woody bevorzuge ich bf2.4. Beide benutzen eine Version von `boot-floppies` zum Erzeugen von Boot Disketten.

Besitzen Sie eine PCMCIA Netzwerkkarte, so müssen Sie die Standard Bootkonfiguration nutzen (dies enthält die größte Anzahl von Disketten und alle Treibermodule) und die Netzwerkkarte im PCMCIA Menü konfigurieren; man darf sie nicht im Standard Netzwerk Dialog einrichten.

Für spezielle Systeme kann es notwendig sein, eine angepasste Rettungsdiskette zu erzeugen. Dies kann durch Ersetzen des Kernelimages namens "linux" auf der Debian Rettungsdiskette, durch einen anderes komprimiertes Kernelimage, das speziell für den Rechner erzeugt wurde, erreicht werden. Details sind in `readme.txt` auf der Rettungsdiskette zu finden. Diese nutzt das MS-DOS Dateisystem, so dass ein beliebiges System zum Auslesen und Editieren verwendet werden kann. Dies sollte für Personen mit spezieller Netzwerkkarte, etc. leicht zu bewältigen sein.

In Sarge wird wohl `debian-installer` und/oder `pgi` zur Erzeugung von Bootdisketten verwendet werden.

3.1.8 Installation

Folgen Sie den offiziellen Anweisungen, die unter `http://www.debian.org/releases/stable/installmanual` oder `http://www.debian.org/releases/testing/installmanual` (befindet sich in Arbeit und ist manchmal eventuell nicht vorhanden) gefunden werden können.

Sollten Sie ein System mittels `boot-floppies` aus der `testing` Distribution installieren, so kann es sein, dass sie während der Installation eine Konsole mittels Alt-F2 öffnen müssen, um in der Datei `/etc/apt/sources.list` manuell Einträge von "stable" nach "testing" zu ändern, um die APT Quellen anzupassen.

Ich tendiere dazu, `lilo` in Partitionen wie `/dev/hda3` zu installieren, während `mbr` in `/dev/hda` installiert wird. Dies minimiert das Risiko, Bootinformationen zu überschreiben.

Während des Installationsvorgangs wähle ich folgende Werte.

- MD5 Passwörter "yes"
- shadow Passwörter "yes"
- Install "advanced" (dselect **) und select
  - Abwahl von emacs (falls gewählt), nvi, tex, telnet, talk(d);
  - Auswahl von mc, vim und entweder nano-tiny oder elvis-tiny. Vergleiche '`dselect`' auf Seite 67. Selbst wenn Sie ein Emacs Anhänger sind, vermeiden Sie

es zunächst und begnügen Sie sich mit nano während der Installation. Man sollte auch andere große Pakete, so wie TeX (Potato wählte dies aus) zu diesem Moment noch nicht installieren. Vergleiche 'Rettungseditoren' auf Seite 161 für den Grund, nano-tiny oder elvis-tiny jetzt zu installieren.

- Alle Konfigurationsfragen während der Installationsdialoge der einzelnen Pakete mit "y" beantworten (ersetzt aktuellen Wert).
- `exim`: wähle 2 für Rechner, da ich meine E-Mails durch den SMTP Server meines Internetproviders (ISP) verschicke.

Für weitere Informationen über dselect vergleiche '`dselect`' auf Seite 67.

3.1.9 Hosts und IP im LAN

Beispiel einer LAN Konfiguration (Klasse C Subnet: 192.168.1.0/24):

```
Internet
   |
   +--- Externer ISP für POP Service (Zugriff mittels fetchmail)
   |
  Access point ISP bietet DHCP Service und SMTP relay service an
   |                    :
  Kabelmodem          (Dialup)
   |                    :
LAN Gateway Maschine, externer Port: eth0 (IP durch ISP's DHCP)
 altes Notebook (IBM Thinkpad, 486 DX2 50MHz, 20MB RAM)
 Linux 2.4 Kernel mit ext3 Dateisystem.
 "ipmasq" Paket (mit umfangreichem Patch, NAT und Firewall)
 "dhcp-client" Paket, konfiguriert für eth0 (überschreibt DNS Werte)
 "dhcp" Paket konfiguriert für eth1
 starte "exim" als smarthost (Mode 2)
 starte "fetchmail" in großen Intervallen (zur Sicherheit)
 starte "bind" als Cache Nameserver fürs Internet vom LAN
              als Nameserver fürs LAN Netzwerk vom LAN
 starte "ssh" auf Port 22 und 8080 (Verbindung von überall)
 starte "squid" als Cache Server für das Debianarchiv (für APT)
LAN Gateway Maschine, interner Port: eth1 (IP = 192.168.1.1, fest)
                        |
         +--- LAN Switch (10 base T) ---+
         |                              |
Einige feste IP Clients im LAN Einige DHCP Clients im LAN
(IP = 192.168.1.2-127, fest)   (IP = 192.168.1.128-200, dynamisch)
```

Man vergleiche 'Aufsetzen eines Gateway mit einem Debian System' auf Seite 155 für die Details zur Konfiguration des LAN Gateway Server.

3.1.10 Benutzerkonten

Um ein konsistentes Verhalten zwischen verschiedenen Systemen zu gewährleisten, sind die ersten paar Benutzerkonten in meinem System stets gleich.

Ich erzeuge immer zuerst ein normales Benutzerkonto mit einem Namen wie “admin” (uid=1000) und leite die E-Mails an root an dies weiter. Diesem Konto wird die Gruppe `adm` zugeordnet (siehe “‘Warum GNU `su` nicht die `wheel` Gruppe unterstützt”’ auf Seite 120) und es können ihm einige root Privilegien mit `su` (PAM benutzend) oder mit `sudo` zugeordnet werden. Vergleiche ‘Hinzufügen eines neuen Nutzers’ auf Seite 50 für Details.

3.1.11 Anlegen von Dateisystemen

Partitionierung der Festplatte

Ich bevorzuge die Verwendung verschiedener Partitionen für verschiedene Verzeichnisbäume, um Schäden bei Systemabstürzen vorzubeugen. Ein Beispiel dafür ist:

```
/          == (/ + /boot + /bin + /sbin)
           == 50MB+
/tmp       == 100MB+
/var       == 100MB+
/home      == 100MB+
/usr       == 700MB+ mit X
/usr/local == 100MB
```

Die Größe des `/usr` Verzeichnis hängt sehr von installierten X-Windows Anwendungen und der Dokumentation ab. `/usr/` kann rund 300MB groß sein, wenn man nur mit der Konsole arbeitet, wobei 2GB–3GB keine unübliche Größe ist, wenn man viele Gnome Anwendungen installiert hat. Wird `/usr/` zu groß, so ist das Verschieben von `/usr/share/` auf eine andere Partition die effektivste Lösung. Mit den neuen großen vorgefertigten Linux 2.4 Kernel, kann `/` mehr als 200MB benötigen.

Zum Beispiel ist der aktuelle Status meines Internet Gateway Rechners wie folgt (Ausgabe des `df -h` Kommandos):

```
Filesystem          Größe Benut  Verf Ben% Eingehängt auf
/dev/hda3            300M  106M  179M  38% /
/dev/hda7            100M   12M   82M  13% /home
/dev/hda8            596M   53M  513M  10% /var
/dev/hda6            100M  834k   94M   1% /var/lib/cvs
/dev/hda9            596M  222M  343M  40% /usr
/dev/hda10           596M  130M  436M  23% /var/cache/apt/archives
/dev/hda11           1.5G  204M  1.2G  14% /var/spool/squid
```

(Der große reservierte Bereich für `/var/spool/squid/` ist für einen Proxy Cache für den Paketdownload.)

Es folgt die `fdisk -l` Ausgabe zur Veranschaulichung einer möglichen Partitionierung:

```
# fdisk -l /dev/hda # Kommentar

/dev/hda1         1      41    309928+   6  FAT16 # DOS
/dev/hda2        42      84    325080   83  Linux # nicht benutzt
/dev/hda3  *     85     126    317520   83  Linux # root Partition
/dev/hda4       127     629   3802680    5  Extended
/dev/hda5       127     143    128488+  82  Linux swap
/dev/hda6       144     157    105808+  83  Linux
/dev/hda7       158     171    105808+  83  Linux
/dev/hda8       172     253    619888+  83  Linux
/dev/hda9       254     335    619888+  83  Linux
/dev/hda10      336     417    619888+  83  Linux
/dev/hda11      418     629   1602688+  83  Linux
```

Einige nicht benutzte Partitionen existieren. Diese sind für die Installation einer zweiten Linux Distribution oder als Erweiterung für wachsende Verzeichnisbäume vorgesehen.

Mounten von Dateisystemen

Das korrekte Mounten der obigen Dateisysteme wird durch folgende `/etc/fstab` erreicht:

```
# /etc/fstab: statische Dateisystem Informationen.
#
# Dateisystem Mountverzeichnis Typ  Optionen     dump Durchgang
/dev/hda3     /              ext2  defaults,errors=remount-ro 0 1
/dev/hda5     none           swap  sw                      0 0
proc          /proc          proc  defaults                0 0
/dev/fd0      /floppy        auto  defaults,user,noauto    0 0
/dev/cdrom    /cdrom       iso9660 defaults,ro,user,noauto 0 0
#
# einige separate Partitionen
/dev/hda7     /home          ext2  defaults                0 2
/dev/hda8     /var           ext2  defaults                0 2
/dev/hda6     /var/lib/cvs   ext2  defaults                0 2
# noatime beschleunigt den Dateizugriff beim Lesen
/dev/hda9     /usr           ext2  defaults,noatime        0 2
/dev/hda10    /var/cache/apt/archives ext2 defaults        0 2

# sehr große Partition für Proxy Cache
```

```
/dev/hda11    /var/spool/squid ext2 rw                    0 2

# DOS Backup
/dev/hda1     /mnt/dos        vfat  rw,noauto             0 0
# Backup des Linux Systems
/dev/hda2     /mnt/linux      ext2  rw,noauto             0 0
#
# nfs mounts
mickey:/      /mnt/mickey     nfs   ro,noauto,intr        0 0
goofy:/       /mnt/goofy      nfs   ro,noauto,intr        0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2
```

Für NFS wird `noauto,intr` kombiniert mit der standardmäßigen `hard` Option benutzt. Damit ist es möglich, aufgrund abgebrochener Verbindung hängende Prozesse mit Strg-C zu beenden.

Für einen mit Samba (smbfs) angebundenen Windows Rechner ist `rw,auto,soft,intr` eine gute Option. Siehe auch 'Samba Konfiguration' auf Seite 42.

Für ein Diskettenlaufwerk ist `noauto,rw,sync,user,exec` zu empfehlen, da dies Beschädigungen des Dateisystems durch einem unbeabsichtigten Diskettenauswurf vor einem umount Aufruf verhindert, allerdings verlangsamt dies auch den Schreibzugriff.

Autofs mount

Wichtige Punkte zu auto mount:

- Laden Sie das `vfat` Modul, so dass `/etc/auto.misc` die Zeile `-fstype=auto` enthalten kann:

```
# modprobe vfat # vor dem Zugriff auf die Diskette
 ... oder um dies zu automatisieren
# cat >>/etc/modules
vfat
^D
 ... jetzt das System neu starten.
```

- Die Datei `/etc/auto.misc` ist wie folgt anzupassen:

```
floppy -fstype=auto,sync,nodev,nosuid,gid=100,umask=000 :/dev/fd0
 ... wobei gid=100 für "users" steht.
```

- Erzeugen Sie nun `cdrom` und `floppy` Links in `/home/`*`user`*, die auf `/var/autofs/misc/cdrom` bzw. `/var/autofs/misc/floppy` verweisen.
- Nun ist *user* der "users" Gruppe hinzuzufügen.

NFS mount

Der externe Linux NFS Server (goofy) befindet sich hinter einem Firewall (gateway). Ich habe eine sehr lockere Sicherheitspolitik in meinem LAN, da ich der einzige Nutzer bin. Zum

Aktivieren des NFS Zugangs muss auf der NFS Serverseite die Datei `/etc/exports` erzeugt werden:

```
# /etc/exports: die Zugriffskontrollliste für Dateisysteme, auf die
#               NFS Clients zugreifen sollen.  Siehe exports(5).
/       (rw,no_root_squash)
```

Dies ist nötig, um den NFS Server zu konfigurieren, nachdem die NFS Server- und Client-Pakete installiert und aktiviert wurden.

Der Einfachheit halber erzeuge ich üblicherweise eine einzige 2GB Partition für eine experimentelle oder schnelle zweite Linuxinstallation. Ich teile wahlweise die swap und `/tmp` Partitionen für diese Installationen. Ein aus mehreren Partitionen bestehendes System ist für diese Verwendung zu kompliziert. Wenn nur ein einfaches Konsolen System benötigt wird, so sind 500MB mehr als ausreichend.

3.1.12 DRAM Speicher Richtlinien

Es folgen einige grobe Anhaltspunkte für DRAM.

```
  4MB:  Absolutes Minimum für den Linux Kernel.
 16MB:  Minimum für ein angemessenes Konsolen System.
 32MB:  Minimum für ein einfaches X System.
 64MB:  Minimum für ein X System mit GNOME/KDE.
128MB:  Empfohlen für komfortables Arbeiten mit X und GNOME/KDE.
256MB (oder mehr): Warum nicht, wenn man es sich leisten kann?
        DRAM ist billig.
```

Verwendung der Bootoption `mem=4m` (oder in lilo.conf `append="mem=4m"`) zeigt, wie sich ein System mit 4MB Speicher verhält. Ein lilo Bootparameter wird für Systeme mit mehr als 64MB Speicher und einem alten BIOS benötigt.

3.1.13 Swap Speicher

Ich verwende die folgenden Richtlinien für Swap:

- Jede Swap-Partition ist < 128MB (bei Verwendung alter 2.0 Kernel), bzw. < 2GB (bei Verwendung aktueller Kernel)
- Insgesamt = entweder das 1- bis 2-fache des installierten RAM oder 128MB bis 2GB wird empfohlen
- Die Swap-Partitionen sind auf verschiedene Festplatten zu verteilen und mit den Optionen `sw,pri=1` in `/etc/fstab` zu mounten. Dies stellt sicher, dass der Kernel ein "striping RAID" der Swap-Partitionen nutzt und einen maximalen Durchsatz erreicht.
- Nach Möglichkeit sollte der mittlere Bereich der Festplatte verwendet werden.

Selbst wenn er niemals benötigt wird, ist etwas Swap Speicher (128MB) empfehlenswert, da das System langsamer wird, bevor es wegen Speichermangel abstürzt.

3.2 Bash Konfiguration

Ich passe folgende Shell Startskripte an meine Gewohnheiten an:

```
/etc/bash.bashrc        wird durch eigene Version ersetzt
/etc/profile            wird beibehalten (bis auf $PS1: \w -> \W)
/etc/skel/.bashrc       wird durch eigene Version ersetzt
/etc/skel/.profile      wird durch eigene Version ersetzt
/etc/skel/.bash_profile wird durch eigene Version ersetzt
~/.bashrc               wird für alle Nutzer ersetzt
~/.profile              wird für alle Nutzer ersetzt
~/.bash_profile         wird für alle Nutzer ersetzt
```

Vergleiche Details dazu in meinen Beispielskripten (`examples/`). Ich bevorzuge ein transparentes System, deshalb setze ich `umask` auf 002 oder 022.

`PATH` wird durch die folgenden Konfigurationsdateien in der Reihenfolge

```
/etc/login.defs  - bevor die Shell PATH setzt
/etc/profile     (startet eventuell /etc/bash.bashrc)
~/.bash_profile  (startet eventuell ~/.bashrc)
```

gesetzt.

3.3 Maus Konfiguration

3.3.1 PS/2 Mäuse

Im Falle einer Maus mit einem PS/2 Anschluss am ATX Motherboard, sollte der Signalfluss wie folgt sein:

```
Maus -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

Hier wurde ein symbolischer Link `/dev/mouse` verwendet, der auf `/dev/gpmdata` verweist, um einige Konfigurationsprogramme zufriedenzustellen und die erneute Konfiguration zu vereinfachen. (Entscheidet man sich beispielsweise dazu, den `gpm` Daemon nicht zu nutzen, so muss der symbolische Link `/dev/mouse` auf `/dev/psaux` geändert werden, nachdem `gpm` entfernt wurde.)

Dieser Signalfluss erlaubt, die Tastatur und die Maus zu entfernen und beim Anstecken mit `gpm` neu zu initialisieren. X wird davon nicht negativ beeinflusst!

Das Protokoll des Signalflusses zwischen `gpm` Ausgabe und X Eingabe kann auf zwei Arten implementiert werden, als spezielles Protokoll wie z.B. “ms3” (serielles Microsoft 3-Tasten

Mausprotokoll) oder "raw" (nutzt das selbe Protokoll wie die angeschlossene Maus) und diese Wahl diktiert die Wahl des in der X Konfiguration benutzten Protokolls.

Ich werde die Konfiguration anhand einer 3-Tasten Logitech (traditionelle Unix-artige) PS/2 Maus demonstrieren und im folgenden Beispiel beschreiben.

Sind Sie einer derjenigen, dessen Graphikkarte nicht durch den neuen X Server Version 4, sondern nur durch Version 3 unterstützt wird (einige ATI 64 Bit Karten), so muss `/etc/X11/X86Config` anstatt von `/etc/X11/X86Config-4` im folgenden konfiguriert werden.

Der ms3 Protokoll Ansatz

```
/etc/gpm.conf             | /etc/X11/X86Config-4
==========================+=====================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Konfigurierte Maus"
repeat_type=ms3           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"   "/dev/gpmdata"
sample_rate=              |  Option     "Protocol" "IntelliMouse"
                          | EndSection
```

Wenn diese Methode gewählt wird, so erfolgt die Anpassung des Maustyps allein in der Datei `gpm.conf` und die X Konfiguration bleibt gleich. Man vergleiche meine Beispielskripte (`examples/`).

Der raw Protokoll Ansatz

```
/etc/gpm.conf             | /etc/X11/X86Config-4
==========================+=====================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=raw           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"   "/dev/mouse"
sample_rate=              |  Option     "Protocol" "MouseManPlusPS/2"
                          | EndSection
```

Wenn diese Methode gewählt wird, so erfolgt die Anpassung des Maustyps in `gpm.conf` sowie in der X Konfiguration.

Wie man andere Mäuse einrichtet

Der `gpm` Gerätetyp `autops2` sollte die meisten PS/2 Mäuse erkennen. Unglücklicherweise funktioniert dies nicht immer und ist in vor-Woody Versionen nicht verfügbar. In diesen Fällen sollte man `ps2` oder `imps2` in `gpm.conf` anstatt von `autops2` probieren. Um die von `gpm` unterstützten Mäuse zu bestimmen, kann man `gpm -t help` starten. Man vergleiche `gpm(8)`.

Wird eine 2-Tasten PS/2 Maus verwendet, so sollte das X Protokoll die Option `Emulate3Buttons` nutzen. Der Unterschied des Protokolls zwischen der 2- und 3-Tasten Maus wird automatisch erkannt und für `gpm` eingerichtet, nachdem die mittlere Maustaste einmalig gedrückt wurde.

Für das X Protokoll mit 'Der raw Protokoll Ansatz' auf der vorherigen Seite oder ohne `gpm` nutzt man:

- `IntelliMouse`: Maus mit seriellem Anschluss (`gpm` Repeater mit "ms3")
- `PS/2`: PS/2 Anschluss Maus (dies immer zuerst ausprobieren)
- `IMPS/2`: beliebige PS/2 Anschluss Mäuse (2, 3 oder Scroll Mäuse, besser)
- `MouseManPlusPS/2`: Logitech PS/2 Anschluss Maus
- `...`

Ausführlicheres ist unter Mouse Support in XFree86 (`http://www.xfree86.org/current/mouse.html`) zu finden.

Eine typische Microsoft Scroll Maus funktioniert am besten mit:

```
/etc/gpm.conf            | /etc/X11/X86Config-4
=========================+====================================
device=/dev/psaux        | Section "InputDevice"
responsiveness=          |  Identifier "Konfigurierte Maus"
repeat_type=raw          |  Driver     "mouse"
type=autops2             |  Option     "CorePointer"
append=""                |  Option     "Device"   "/dev/mouse"
sample_rate=             |  Option     "Protocol" "IMPS/2"
                         |  Option     "Buttons" "5"
                         |  Option     "ZAxisMapping" "4 5"
                         | EndSection
```

Für einige aktuelle dünne Toshiba Notebooks, kann das Aktivieren von `gpm` vor PCMCIA im System-V Init Skript helfen, wenn das System sonst hängenbleibt. Eigenartig aber wahr.

3.3.2 USB Mäuse

Stellen Sie sicher, dass Sie alle benötigten Kernel Elemente einkompiliert oder als Modul vorliegen haben:

- Unter "Input core support":
  - "Input core support" (CONFIG_INPUT, `input.o`),

  - "Mouse support" (CONFIG_INPUT_MOUSEDEV, `mousedev.o`),
- Unter "USB support":
  - "Support for USB" (CONFIG_USB, `usbcore.o`),
  - "Preliminary USB device filesystem" (CONFIG_USB_DEVICEFS),
  - "UHCI" oder "OHCI" (CONFIG_USB_UHCI || CONFIG_USB_UHCI_ALT || CONFIG_USB_OHCI, `usb-uhci.o` || `uhci.o` || `usb-ohci.o`),
  - "USB Human Interface Device (full HID) support" (CONFIG_USB_HID, `hid.o`) und
  - "HID input layer support" (CONFIG_USB_HIDINPUT)

Die kleingeschriebenen Namen sind Modulnamen.

Wird devfs nicht genutzt, so ist eine Gerätedatei `/dev/input/mice` mit major 13 und minor 63 wie folgt anzulegen:

```
# cd /dev
# mkdir input
# mknod input/mice c 13 63
```

Für typische **USB** Scroll Mäuse, ist folgende Konfiguration zu empfehlen:

```
/etc/gpm.conf             | /etc/X11/X86Config-4
==========================+=====================================
device=/dev/input/mice    | Section "InputDevice"
responsiveness=           |  Identifier "Konfigurierte Maus"
repeat_type=raw           |  Driver     "mouse"
type=autops2              |  Option     "SendCoreEvents" "true"
append=""                 |  Option     "Device"   "/dev/input/mice"
sample_rate=              |  Option     "Protocol" "IMPS/2"
                          |  Option     "Buttons" "5"
                          |  Option     "ZAxisMapping" "4 5"
                          | EndSection
```

Man vergleiche das Linux USB Projekt (`http://www.linux-usb.org/`) für weitere Informationen.

3.3.3 Touchpad

Obwohl das Touchpad eines Laptops standardmäßig eine 2-Tasten PS/2 Maus emuliert, ermöglicht das `tpconfig` Paket die volle Kontrolle über das Gerät. Die Einstellung `OPTIONS="--tapmode=0"` in `/etc/default/tpconfig` deaktiviert z.B. das ärgerliche "klicken beim Antippen" Verhalten. Folgende `/etc/gpm.conf` ermöglicht das Nutzen des Touchpads und einer externen USB Maus in der Konsole:

```
device=/dev/psaux
```

```
responsiveness=
repeat_type=ms3
type=autops2
append="-M -m /dev/input/mice -t autops2"
sample_rate=
```

3.4 NFS Konfiguration

NFS wird durch `/etc/exports` eingerichtet:

```
# echo "/ *.domainname-für-lan-hosts(rw,no_root_squash,nohide)" \
        >> /etc/exports
```

Man vergleiche meine Beispielskripte für Details (`examples/`).

3.5 Samba Konfiguration

Referenzen:

- `http://www.samba.org/`
- `samba-doc` Paket

Das Einrichten von Samba im "share" Mode ist viel einfacher, da dies die Laufwerke wie unter WfW anspricht. Aber es wird empfohlen, den "user" Mode zu verwenden.

Samba kann durch `debconf` oder `vi` konfiguriert werden:

```
# dpkg-reconfigure --priority=low samba # in Woody
# vi /etc/samba/smb.conf
```

Man vergleiche meine Beispielskripte für Details (`examples/`).

Das Hinzufügen eines neuen Nutzers zur `smbpasswd` Datei kann mit `smbpasswd` erfolgen:

```
$ su -c "smbpasswd -a Benutzername"
```

Man sollte sicherstellen, dass man verschlüsselte Passwörter für optimale Kompatibilität verwendet.

Der Wert `os level` ist entsprechend zu den folgenden Systemäquivalenzen zu setzen (je höher der Wert, um so größer ist die Priorität als Server):

```
0:      Samba mit geringer Priorität
1:      WfW 3.1, Win95, Win98, Win/Me?
16:     Win NT WS 3.51
17:     Win NT WS 4.0
32:     Win NT SVR 3.51
33:     Win NT SVR 4.0
255:    Samba mit hoher Priorität
```

Es muss sichergestellt werden, dass die Nutzer der Gruppe angehören, welche dem Verzeichnis zugeordnet ist, das den gemeinsamen Zugriff ermöglicht. Außerdem muss der Verzeichnispfad entsprechende Ausführungsrechte haben, um den Zugriff zu ermöglichen.

3.6 Drucker Konfiguration

Die traditionelle Methode ist `lpr/lpd`. Es gibt allerdings ein neues System: CUPS™ (Common UNIX Printing System). PDQ ist ein anderer Ansatz. Man vergleiche Linux Printing HOWTO (`http://www.tldp.org/HOWTO/Printing-HOWTO.html`) für weitere Informationen.

3.6.1 `lpr/lpd`

Für die `lpr/lpd` artigen Spooler (`lpr`, `lprng` und `gnulpr`) ist `/etc/printcap` wie folgt für PostScript- und Nur-Text-Drucker anzupassen (die Grundlagen):

```
lp|alias:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:
```

Bedeutung der obigen Zeilen:

- Kopfzeile: *lp* – Name der Druckerwarteschlange, *alias* = alias
- mx#0 – maximale Dateigröße ist nicht limitiert
- sh – Unterdrücken eines Deckblatts
- lp=/dev/lp0 – lokale Druckerschnittstelle oder port@host für entfernte Drucker

Dies ist eine gute Konfiguration für einen PostScript Drucker. Auch für das Drucken von einem Windows Rechner mittels Samba ist dies eine geeignete Konfiguration für alle von Windows unterstützten Drucker (bidirektionale Kommunikation wird nicht unterstützt). Man muss die entsprechende Druckerkonfiguration auf dem Windows Rechner nutzen.

Sollte kein PostScript Drucker vorhanden sein, muss man einen Filter mittels `gs` aktivieren. Es gibt viele Konfigurationstools für das Einrichten von `/etc/printcap`. Jede der folgenden Möglichkeiten kommt in Frage:

- `gnulpr`, (`lpr-ppd`) und `printtool` – Ich nutze dies.

- `lpr` und `apsfilter`
- `lpr` und `magicfilter`
- `lprng` und `lprngtool`
- `lprng` und `apsfilter`
- `lprng` und `magicfilter`

Um graphische Konfigurationstools wie `printtool` zu verwenden, vergleiche man 'Administratorrechte unter X' auf Seite 138 um root Rechte zu erhalten. Mit `printtool` erzeugte Druckerwarteschlangen nutzen `gs` und agieren wie PostScript Drucker. Beim Zugriff darauf, müssen deshalb PostScript Druckertreiber verwendet werden. Auf der Windows Seite ist "Apple LaserWriter" der Standard.

3.6.2 CUPS™

Das Common UNIX Printing System (oder CUPS™) wird wie folgt installiert:

```
# apt-get install cupsys foomatic-bin foomatic-db
# apt-get install cupsys-bsd cupsys-driver-gimpprint
```

Danach muss das System mit einem Web Browser konfiguriert werden:

```
$ meinBrowser http://localhost:631
```

Ein Drucker kann wie folgt hinzugefügt werden:

- klicken auf "Printers" auf der Hauptseite und danach "Add Printer",
- Eingabe von "root" für den Nutzernamen und des entsprechenden Passworts,
- folgen Sie den weiteren Anweisungen zum Hinzufügen des Druckers,
- gehen Sie zurück zur "Printers" Seite und klicken Sie "Configure Printer" und
- fahren Sie fort, die Papiergröße, Auflösung und andere Parameter zu spezifizieren.

Weitere Informationen sind unter `http://localhost:631/documentation.html` und `http://www.cups.org/cups-help.html` zu finden.

Für einen 2.4 Kernel vergleiche man auch 'Parallel Port Unterstützung' auf Seite 93.

3.7 Andere Installations Hinweise

3.7.1 Installation einiger weiterer Pakete nach der Basisinstallation

Sind Sie den bisherigen Anweisungen gefolgt, so haben Sie ein kleines aber funktionsfähiges Debian System. Es ist nun ein guter Zeitpunkt, um größere Pakete zu installieren.

- Man starte `tasksel`. Vergleiche '*Aufgaben-Profile* mit `tasksel` oder `aptitude` installieren' auf Seite 66.

  Man kann folgendes bei Bedarf wählen:
  - End-user – X Window System
  - Development – C and C++
  - Development – Python
  - Development – Tcl/Tk
  - Miscellaneous – TeX/LaTeX environment
  - Ich nutze `tasksel` auch oft, um mir einen Überblick über die einzelnen Komponenten eines Menüpunktes mittels <Task Info> zu verschaffen, und diese einzeln mittels `dselect` zu installieren.

- Starten von `dselect`.

  Das erste was man jetzt wohl tun wird, ist den favorisierten Editor und alle Programme die man benötigt, auszuwählen. Man kann mehrere Emacs Varianten gleichzeitig installieren. Man vergleiche '`dselect`' auf Seite 67 und 'Populäre Editoren' auf Seite 161.

  Man kann auch einige der Standardprogramme durch Programme mit größerer Funktionalität ersetzen.
  - lynx-ssh (anstatt lynx)
  - ...

- ...

Ich editiere gewöhnlich `/etc/inittab` zum einfacheren Herunterfahren des Rechners.

```
...
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -h now
...
```

3.7.2 Module

Module für Gerätetreiber werden während der Basisinstallation konfiguriert. `modconf` bietet danach eine menübasierte Modulkonfiguration. Dieses Programm ist sehr nützlich, wenn einige Module während der Basisinstallation ausgelassen wurden oder ein neuer Kernel seitdem installiert wurde.

Alle vorzuladenden Module müssen in `/etc/modules` aufgelistet werden. Ich nutze auch `lsmod` und `depmod` um dies manuell zu kontrollieren.

Man sollte auch sicherstellen, einige Zeilen zu `/etc/modules` hinzuzufügen, um IP Masquerading (FTP, etc.) für 2.4 Kernel zu handhaben. Man vergleiche 'Der modularisierte Kernel 2.4' auf Seite 89, speziell 'Netzwerk Funktionalität' auf Seite 91.

3.7.3 Grundlegende CD-RW Einrichtung

Editieren Sie die folgenden Dateien:

```
/etc/lilo.conf  (append="hdc=ide-scsi ignore=hdc" ist hinzuzufügen und
                 lilo zum Aktivieren zu starten)
/dev/cdrom      (symbolischer Link # cd /dev; ln -sf scd0 cdrom)
/etc/modules    (hinzufügen von "ide-scsi" und "sg", sowie "sr" danach wenn
                 nötig)
```

Man vergleiche 'CD-Brenner' auf Seite 122 für Details.

3.7.4 Großer Speicher und automatisches Abschalten

Editieren Sie `/etc/lilo.conf` wie folgt, um Bootparameter für großen Speicher (für 2.2 Kernel) und automatisches Abschalten (für APM) zu setzen:

```
append="mem=128M apm=on apm=power-off noapic"
```

Nun muss noch `lilo` zum Aktivieren dieser Einstellungen gestartet werden. `apm=power-off` wird für SMP Kernel verwendet und `noapic` ist für meine fehlerhafte SMP-Hardware nötig. Das selbe kann direkt durch Angabe der Optionen beim Boot-Prompt erreicht werden. Man vergleiche 'Weitere Tricks mit dem Boot-Prompt' auf Seite 98.

Wurde APM als Modul kompiliert wie in den Standard Debian 2.4 Kernel, so ist `insmod apm power_off=1` nach dem Booten zu starten oder `/etc/modules` wie folgt zu modifizieren:

```
# echo "apm power_off=1" >>/etc/modules
```

Alternativ kann mit ACPI Unterstützung das selbe mit neueren Kernel erreicht werden und dies scheint SMP freundlicher zu sein (erfordert ein neueres Motherboard). Der 2.4 Kernel auf neueren Motherboards sollte großen Speicher korrekt erkennen.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

auch die folgenden Zeilen sollten in `/etc/modules` in dieser Reihenfolge hinzugefügt werden:

```
ospm_busmgr
ospm_system
```

Es ist auch möglich, den Kernel mit all diesen auf "y" gesetzten Kerneloptionen zu kompilieren. Auf jeden Fall sind keine Bootparameter mit ACPI nötig.

3.7.5 Eigenartige Probleme beim Zugriff auf einige Webseiten

Aktuelle Linux Kernel aktivieren standardmäßig ECN, was Zugriffsprobleme bei einigen Webseiten auf schlechten Routern zur Folge haben kann. Überprüfen des ECN Status:

```
# cat /proc/sys/net/ipv4/tcp_ecn
 ... oder
# sysctl net.ipv4.tcp_ecn
```

Zum Ausschalten ist

```
# echo "0" > /proc/sys/net/ipv4/tcp_ecn
 ... oder
# sysctl -w net.ipv4.tcp_ecn=0
```

zu verwenden. Um TCP ECN bei jedem Start zu deaktivieren, ist `/etc/sysctl.conf` zu editieren und

```
net.ipv4.tcp_ecn = 0
```

hinzuzufügen.

3.7.6 Dialup PPP Konfiguration

Das `pppconfig` Paket ist zur Nutzung eines dialup (Modem) PPP Zugangs zu installieren.

```
# apt-get install pppconfig
# pppconfig
 ... folgen Sie den Anweisungen zur dialup PPP Konfiguration
# adduser user_name dip
 ... erlaube user_name Zugriff auf dialup PPP
```

Dialup PPP Zugriff kann durch den Nutzer (*user_name*) mit

```
$ pon ISP_name  # starte PPP Zugang zum ISP
 ... erfreuen Sie sich des Internets
$ poff ISP_name # beende PPP Zugang, ISP_name optional
```

gestartet werden. Man vergleiche `/usr/share/doc/ppp/README.Debian.gz` für mehr Details.

Alternativ kann das `wvdial` Paket zum dialup PPP Zugang genutzt werden. Beachten Sie bitte den bekannten Fehler http://bugs.debian.org/82095, der manchmal das Einwählen für nicht-root Nutzer verhindert.

Alle Wählprogramme nutzen den `pppd` Daemon, der Programme aus `/etc/ppp/ip-up.d/` und `/etc/ppp/ip-down.d/` startet, nachdem die Verbindung auf- bzw. abgebaut wurde. Dies wird genutzt, um Mails zu empfangen und zu versenden.

3.7.7 Andere Konfigurationsdateien in `/etc/`

Da in der Standard Debian Installation `/etc/cron.deny` fehlt, sollte diese erzeugt werden (man kann `/etc/at.deny` kopieren).

Kapitel 4

Debian Tutorials

Dieser Abschnitt bietet eine grobe Orientierung zur Linuxwelt für den Neueinsteiger. Sollten Sie Linux schon eine Weile nutzen, so können Sie Ihr Wissen testen.

4.1 Informationsquellen

Die ultimativen Referenzen für Debian sind im Debian Dokumentations Projekt (DDP) (`http://www.debian.org/doc/`) zu finden. Viele dieser Dokumente werden in `/usr/share/doc/` installiert. Man sollte auch in `/usr/share/doc-base/` schauen, dies enthält Verweise auf die Dokumentation auf dem System. Man kann `export CDPATH=.:/usr/share/doc:/usr/src/local` zu `~/.bash_profile` hinzufügen, um einfacher auf die Dokumentationsverzeichnisse zuzugreifen.

Das Linux Dokumentations Projekt (LDP) (`http://www.tldp.org/`) hat die maßgeblichsten allgemeinen Linux Referenzen. Der LDP Inhalt wird gewöhnlich in `/usr/share/doc/HOWTO/` installiert.

Man kann durch Dokumente auf lokalen und entfernten FTP Servern mittels F9 im Midnight Commander (vergleiche 'Midnight Commander (MC)' auf Seite 53) navigieren.

4.2 Die Linux Konsole

4.2.1 Login

In einem gewöhnlichen Linux System gibt es sechs unabhängige Pseudo-Terminals. Man kann zwischen diesen durch gleichzeitiges Drücken von Alt und F1–F6 wechseln. Jedes Pseudo-Terminal erlaubt ein unabhängiges Einloggen. Das Mehrbenutzersystem ist eine großartige Unix Eigenschaft und sehr suchterzeugend.

Es ist unter Unix eine gute Angewohnheit, sich für die meisten Arbeiten als normaler Nutzer anzumelden. Ich muss zugeben, dass ich den Administrationszugang (root) öfters als nötig der Einfachheit halber und aus Schlamperei verwendete.

Nun verwende ich regelmäßig einen gewöhnlichen Zugang mit den Kommandos `sudo`, `super` oder `su -c` um begrenzten root Zugang zu erhalten.

4.2.2 Hinzufügen eines neuen Nutzers

Nach der Systeminstallation richte ich gewöhnlich einen normalen Nutzerzugang ein. Ist der Nutzername "pinguin", so wird

```
# adduser pinguin
```

ihn erzeugen.

Ich nutze das `vigr` Kommando, um `/etc/group` wie folgt zu editieren:

```
adm:x:4:admin
src:x:40:admin, debian, ...
...
```

Im standardmäßig installierten System, können Mitglieder der `adm` Gruppe viele Logdateien in `/var/log/` lesen und `xconsole` nutzen. Der `staff` Gruppe gehört `/home` an, was den Mitgliedern das Verwalten von Nutzerzugängen erlaubt. Der `src` Gruppe gehört `/usr/src /` an, was zur Kernelkompilation, etc. genutzt wird. Für die offizielle Bedeutung von Nutzern und Gruppen vergleichen Sie Nutzer und Gruppen (`/usr/share/doc/base-passwd/users-and-groups.html`).

Ich persönlich verwende die `adm` Gruppe für Nutzer, die administrative Aufgaben erfüllen und das exklusive `su` Privileg haben (vergleichen Sie "'Warum GNU `su` nicht die `wheel` Gruppe unterstützt"' auf Seite 120) und `src` für CVS (siehe 'CVS' auf Seite 167).

Überprüfen Sie `adduser`, `addgroup`, `vipw`, `vipw -s`, `vigr` und `vigr -s` für die korrekte Konfiguration von Nutzern und Gruppen.

4.2.3 Wie man den Rechner herunterfährt

Wie nahezu jedes moderne Betriebssystem, das Dateien im Speicher zwischenspeichert, muss Linux richtig heruntergefahren werden, bevor der Rechner ausgeschaltet werden kann. Das entsprechende Kommando im Mehrbenutzerbetrieb ist:

```
# shutdown -h now
```

Im Einzelnutzermodus ist

```
# poweroff -i -f
```

zu verwenden. Warten Sie bis das System "System halted" ausgibt und schalten Sie danach aus. Wenn APM im BIOS und Linux aktiviert ist, schaltet sich der Rechner selbstständig aus. Man vergleiche 'Großer Speicher und automatisches Abschalten' auf Seite 46 für Details.

4.2.4 Kommandozeilen basiertes Editieren

Die Standard-Shell `bash` besitzt einige Fähigkeiten zur Bearbeitung von alten Eingaben (der History). Benutzen Sie einfach die Cursor-auf Taste um alte Eingaben zurückzuholen und danach wie üblich die Cursor Tasten. Andere wichtige Tastenkombinationen sind:

```
Strg-U:             Lösche die Zeile bis vor den Cursor
Strg-D:             Beende die Eingabe
markieren mit linker Maustaste:
                    Auswahl in die Zwischenablage kopieren (gpm)
mittlere Maustaste: Einfügen der Zwischenablage an der
                    Cursorposition (gpm)
Strg-V TAB:         Eingabe eines Tabulators ohne Expandierung.
```

In einer normalen Linux Konsole funktioniert nur die linke Strg Taste wie erwartet.

4.2.5 Kommandoausführung

Ein typisches Kommando wird durch die folgende Shell Zeile ausgeführt:

```
$ LC_ALL=de ls -la
```

Hier wird das Programm `ls` im Vordergrund ausgeführt, mit der für Deutsch auf `de` gesetzten Umgebungsvariablen `LC_ALL`. `-la` ist das Kommandozeilen Argument und bewirkt eine detaillierte Ausgabe. Wird die Kommandozeile mit dem & Zeichen abgeschlossen, so wird das Kommando im Hintergrund ausgeführt. Dies ermöglicht es Nutzern, mehrere Programme in einer einzigen Shell zu starten.

Die Ausführung des Kommandos kann durch folgende Tastenkombinationen beeinflusst werden.

```
Strg-C:          Beenden des Programms
Strg-Z:          Temporäres Unterbrechen des Programms
Strg-S:          Unterbrechen der Bildschirmausgabe
Strg-Q:          Ausgabe wieder aktivieren
Strg-Alt-Entf:   Neustarten/Beenden des Systems (siehe /etc/inittab)
```

Für das Management der Programmausführung vergleiche `bash(1)` für `jobs`, `fg`, `bg` und `stop`.

4.2.6 Die grundlegensten Kommandos zur Erinnerung

Die folgenden Kommandos sind für Unix Systeme fundamental:

```
ls, ls -al, ls -d, pwd, cd, cd ~Benutzer, cd -,
cat /etc/passwd, less, bg, fg, kill, killall,
uname -a, type Kommandoname, sync, netstat,
ping, traceroute, top, vi, ps aux, tar, zcat,
grep, ifconfig, ...
```

Sie können mehr über diese Kommandos herausfinden, indem Sie sie in der Kommandozeile oder nach `man` bzw. `info` eingeben. Viele Linux Kommandos geben eine kurze Meldung zur Benutzung aus, wenn sie mit einer der folgenden Methoden gestartet werden:

```
$ Kommandoname --help
$ Kommandoname -h
```

`whatis` *`Kommandoname`* gibt eine einzeilige Zusammenfassung zu jedem Kommando mit einer Manual-Page auf dem System aus.

4.2.7 Das X Window System

Das X Window System lässt sich von der Konsole mit

```
# exec startx
```

starten. Ein Rechtsklick auf den Hintergrund bringt eine Menüauswahl zum Vorschein.

4.2.8 Wichtige Tastaturkommandos

Einige bedeutende Tastenkombinationen für die Linux Konsole sind ("Plus", "Minus" beziehen sich auf den numerischen Ziffernblock):

```
Alt-F1 bis F6:       Wechsel auf anderes Pseudo-Terminal
Strg-Alt-F1 bis F6:  Wechsel auf anderes Pseudo-Terminal
                     (aus X-Windows, DOSEMU, etc.)
Alt-F7:              Wechsel zu X-Windows
Strg-Alt-Minus:      Ändern der Auflösung in X-Windows
Strg-Alt-Plus:       Ändern der Auflösung in X-Windows (umgekehrte
                     Reihenfolge)
Strg-Alt-Rücktaste:  Beenden des X11 Server-Programms
Alt-X, Alt-C, Alt-V: In üblichen Windows/Mac Tastenkombinationen
                     zum Ausschneiden, Kopieren und Einfügen wird
                     die Strg Taste durch Alt in einigen
                     Programmen wie Netscape Composer ersetzt.
```

4.3 Midnight Commander (MC)

Der Midnight Commander (MC) ist ein GNU "Schweizer Armeemesser" für die Linux Konsole und andere textbasierte Umgebungen.

4.3.1 Installation von MC

```
# apt-get install mc
```

Danach sollten Sie `~/.bashrc` (oder `/etc/bash.bashrc`, was von `.bashrc` verwendet wird) entsprechend der Manual-Page `mc(1)` unter `-P` ändern. Dies bewirkt eine Änderung des Arbeitsverzeichnisses von MC beim Beenden.

Befindet man sich in einem Terminal wie z.B. `kon` oder `Kterm` für japanische Unterstützung, das verschiedene graphische Zeichen unterstützt, so hilft eventuell das Hinzufügen von `-a` zu MC's Kommandozeile bei Problemen.

4.3.2 Starten von MC

```
$ mc
```

MC bietet für alle Dateioperationen Menüpunkte an, so dass nur minimaler Aufwand nötig ist.

4.3.3 Dateimanager

Standardmäßig gibt es zwei Verzeichnisansichten mit Dateilisten. Ein anderer nützlicher Modus ist das Setzen des rechten Fensters auf "Info", um Dateizugriffsrechte und andere Informationen zu sehen. Es folgen einige wichtige Tastenkombinationen. Mit einem laufenden `gpm` Daemon kann man auch die Maus nutzen. (Stellen Sie sicher, dass Sie die Shift Taste drücken, um das normale Verhalten von Entfernen und Einfügen im MC zu erhalten.)

- F1: Hilfe
- F3: interner Dateibetrachter
- F4: interner Editor
- F9: aktiviert das Pulldown Menü
- F10: beendet den Midnight Commander
- Tab: wechseln zwischen den beiden Fenstern
- Einf: markieren einer Datei für eine Operation wie Kopieren
- Entf: löschen einer Datei (Seien Sie vorsichtig – setzen Sie MC auf sicheres Löschen.)
- Cursor Tasten: selbsterklärend

4.3.4 Kommandozeilen Tricks

- Ein `cd` wechselt das Verzeichnis im dargestellten Fenster.
- Strg-Enter oder Alt-Enter kopiert den Dateinamen in die Kommandozeile. Dies kann mit dem `cp` oder `mv` Programm und dem Kommandozeilen Editor genutzt werden.
- Alt-Tab bestimmt die Dateinamenvervollständigung in der Shell.
- Man kann das Startverzeichnis für beide Fenster als Argumente an MC übergeben, z.B. `mc /etc /root`.
- Esc + *Ziffer* == F*n* (d.h. Esc + '1' = F1, etc.; Esc + '0' = F10).
- Esc == Alt (= Meta, M-), d.h. Esc + 'c' steht für Alt-c.

4.3.5 Editor

Der interne Editor besitzt ein interessantes Entfernen-und-Einfügen Schema. Drücken von F3 markiert den Start einer Auswahl, ein zweites F3 markiert das Ende der Auswahl und hebt diese hervor. Danach kann man den Cursor bewegen. Beim Betätigen von F6 wird der hervorgehobene Bereich an die Cursorposition verschoben. Drückt man F5, so wird der ausgewählte Bereich an die aktuelle Cursorposition kopiert und eingefügt. F2 sichert die Datei und F10 beendet den Editor. Die meisten Cursortasten funktionieren wie üblich.

Dieser Editor kann direkt mit einer Datei aufgerufen werden:

```
$ mc -e Datei
$ mcedit Datei
```

Es handelt sich nicht um einen Mehr-Fenster-Editor, aber man kann mehrere Linux Konsolen nutzen, um den selben Effekt zu erreichen. Um zwischen Fenstern zu kopieren, sind die Alt-F*n* Tasten zum Wechseln der virtuellen Konsole und "Datei->Datei einfügen" oder "Datei->In Datei kopieren" zum Verschieben eines Teils einer Datei in eine andere zu verwenden.

Dieser interne Editor kann durch einen beliebig anderen externen Editor ersetzt werden.

Viele Programme nutzen die Umgebungsvariable `EDITOR` oder `VISUAL` um zu bestimmen, welcher Editor zu verwenden ist. Sollten Sie mit `vim` nicht vertraut sein, setzen Sie diese auf `mcedit`, indem Sie folgende Zeilen zu `~/.bashrc` hinzufügen:

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

Ich empfehle, diese wenn möglich auf `vim` zu setzen. Die Verwendung der vi(m) Kommandos ist das richtige Vorgehen, da sie stets in einer Linux/Unix Umgebung zu finden sind.

4.3.6 Dateibetrachter

Dies ist ein guter Viewer und prima dazu geeignet, Wörter in Dokumenten zu suchen. Ich nutze dies immer für Dateien im `/usr/share/doc` Verzeichnis. Dies ist der schnellste Weg, um sich einen Überblick über die mehr als umfangreiche Linux Dokumentation zu verschaffen. Der Dateibetrachter kann direkt mittels

```
$ mc -v Datei
```

gestartet werden. (Es ist zu beachten, dass einige Pakete die Regeln verletzen und ihre Dokumentation noch unter `/usr/doc` ablegen.)

4.3.7 Auto-Start Eigenschaften

Drückt man Enter für eine Datei, so wird ein entsprechendes Programm den Inhalt der Datei verarbeiten. Dies ist eine sehr nützliche Eigenschaft von MC.

```
ausführbare Dateien:  Kommando ausführen
man, html Dateien:    Darstellen des Inhalts mit Dateibetrachter
tar, gz, rpm Dateien: Inhalt als Unterverzeichnis einblenden
```

Damit diese Eigenschaften funktionieren, dürfen betrachtbare Dateien nicht ausführbar sein. Ihr Status kann mittels des `chmod` Kommandos und dem Dateimenü von MC geändert werden.

4.3.8 Virtuelles FTP Dateisystem

MC kann zum Zugriff auf Dateien übers Internet mittels FTP genutzt werden. Aktivieren Sie mittels F9 das Menü und drücken Sie dann 'p' für das virtuelle FTP Dateisystem. Eine URL wird in der Form `Benutzername:Passwort@hostname.domainname` eingegeben und MC stellt daraufhin das entfernte Verzeichnis wie ein lokales dar.

4.4 Grundlagen des GNU/Linux Dateisystems

Jede Datei und jedes Verzeichnis eines GNU/Linux Systems ist mit einem Nutzer verknüpft, der sie/es besitzt und einer Gruppe, zu welcher sie/es gehört. Alle Dateiinformationen werden in einem **Inode** genannten Datensatz gespeichert.

4.4.1 Zugriffsrechte von Dateien und Verzeichnissen

Die Zugriffsrechte von Dateien und Verzeichnissen sind separat für die folgenden drei Nutzerkategorien definiert:

- den **Nutzer** (user), der die Datei besitzt (u),
- andere Nutzer der **Gruppe** (group), zu welcher die Datei gehört (g) und
- alle **anderen** (other) Nutzer (o).

Bei Dateien sind je folgende Rechte möglich:

- **lesbar** (readable) (r): zum Auslesen des Inhalts der Datei
- **schreibbar** (writable) (w): zum Modifizieren der Datei
- **ausführbar** (executable) (x): zum Starten der Datei als Kommando

Für Verzeichnisse sind je folgende Rechte möglich:

- **lesbar** (readable) (r): zum Anzeigen des Verzeichnisinhalts
- **schreibbar** (writable) (w): zum Hinzufügen oder Entfernen von Dateien im Verzeichnis
- **ausführbar** (executable) (x): zum Zugriff auf Dateien im Verzeichnis

Das **ausführbar** Recht des Verzeichnisses bezieht sich nicht nur auf das Lesen von Dateien im entsprechenden Verzeichnis, sondern auch auf das Bestimmen der Attribute wie Größe und Modifizierungszeit.

Um diese und einige andere Informationen zu Dateien und Verzeichnissen zu erhalten, kann `ls` genutzt werden. Vergleiche `ls(1)`. Wird `ls` mit der `-l` Option gestartet, so werden die folgenden Informationen in der folgenden Reihenfolge angezeigt:

- der **Dateityp** (erstes Zeichen)
  - -: normale Datei
  - d: Verzeichnis (directory)
  - l: symbolischer Link
  - c: zeichenorientiertes Gerät
  - b: blockorientiertes Gerät
- die **Zugriffsrechte** der Datei (die nächsten neun Zeichen sind je drei Zeichen für Nutzer, Gruppe und Andere in dieser Reihenfolge)
- die **Anzahl der harten Links** zur Datei
- der Name des **Nutzers** der die Datei besitzt
- der Name der **Gruppe** zu der die Datei gehört
- die **Größe** der Datei in Bytes
- **das Datum und die Zeit** der Datei (Modifizierungszeit, mtime)
- der **Name** der Datei.

Um den Besitzer einer Datei zu ändern, wird `chown` von root genutzt. Vergleiche `chown(1)`. Zum Ändern der Gruppe einer Datei, wird `chgrp` vom Besitzer der Datei oder root verwendet. Vergleiche `chgrp(1)`. Das Ändern der Zugriffsrechte von Dateien oder Verzeichnissen erfolgt mit `chmod` durch den Dateibesitzer oder root. Vergleiche `chmod(1)`.

Um z.B. einen Verzeichnisbaum dem Nutzer *nutzer* und der Gruppe *gruppe* zuzuordnen, ist als root folgendes auszuführen:

```
# cd /Pfad/zum/Verzeichnis
```

```
# chown -R nutzer:gruppe .
# chmod -R ug+rwX,o=rX .
```

Es gibt drei weitere spezielle Zugriffsbits:
- **set user ID** (s oder S anstatt Nutzer's x),
- **set group ID** (s oder S anstatt Gruppe's x) und
- **Sticky Bit** (t oder T anstatt Andere's x).

Hier werden in der Ausgabe von `ls -l` diese Bits großgeschrieben, wenn die ausführbar-Bits, die in der Ausgabe nicht mit dargestellt werden können, nicht gesetzt sind.

Das Setzen der **set user ID** einer ausführbaren Datei, erlaubt es einem Nutzer, die Datei mit den Rechten und der ID des Besitzers der Datei auszuführen (z.B. **root**). Ähnlich erlaubt das Setzen des **set group ID** Rechts einer ausführbaren Datei, dass diese mit der Gruppen ID und den Gruppenrechten der Datei (z.B. **root**) gestartet wird. Dies kann zu Sicherheitsproblemen führen, deshalb erfordert das Setzen dieser Bits spezielle Vorsicht.

Setzen der **set group ID** eines Verzeichnisses aktiviert das BSD-artige Dateierzeugungs Schema, bei welchem alle im Verzeichnis erstellten Dateien der Gruppe **group** des Verzeichnisses angehören.

Das Setzen des **Sticky Bits** eines Verzeichnisses verhindert das Entfernen einer Datei im Verzeichnis durch einen Nutzer, der nicht der Besitzer ist. Um den Inhalt einer Datei in einem für alle oder auch für Gruppenmitglieder schreibbaren Verzeichnis wie `/tmp` zu sichern, genügt es nicht das **schreibbar** Recht der Datei zu entfernen, sondern es muss auch das **Sticky Bit** des Verzeichnisses gesetzt werden. Andernfalls kann die Datei entfernt werden und eine neue Datei mit gleichem Namen durch andere Nutzer mit Schreibrechten im Verzeichnis angelegt werden.

Es folgen einige interessante Beispiele für Zugriffsrechte von Dateien.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/pppd
crw-rw----    1 root    dip   108,   0 Jan 18 13:32 /dev/ppp
-rw-r--r--    1 root    root      1051 Jan 26 08:29 /etc/passwd
-rw-r-----    1 root    shadow     746 Jan 26 08:29 /etc/shadow
-rwsr-xr--    1 root    dip     234504 Nov 24 03:58 /usr/sbin/pppd
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt    4 root    root      4096 Feb  9 16:35 /tmp
drwxrwsr-x   10 root    staff     4096 Jan 18 13:31 /usr/local
drwxrwsr-x    3 root    src       4096 Jan 19 08:36 /usr/src
drwxrwsr-x    2 root    mail      4096 Feb  2 22:19 /var/mail
drwxrwxrwt    3 root    root      4096 Jan 25 02:48 /var/tmp
```

4.4.2 Zeitmarken

Es gibt drei Arten von Zeitmarken für GNU/Linux Dateien:
- **mtime**: die Modifizierungszeit (`ls -l`),

- **ctime**: die Zeit der letzten Statusänderung (`ls -lc`) und
- **atime**: die letzte Zugriffszeit (`ls -lu`).

Es ist zu beachten, dass **ctime** nicht der Zeitpunkt der Dateierzeugung ist.

- Das Überschreiben einer Datei ändert sowohl **mtime**, **ctime** als auch **atime** der Datei.
- Das Ändern der Rechte oder des Besitzers einer Datei ändert **ctime** und **atime** der Datei.
- Das Lesen einer Datei ändert **atime** der Datei.

Beachten Sie, dass selbst ein einfacher Lesezugriff auf eine Datei in einem GNU/Linux System normalerweise eine Schreiboperation erfordert, um die **atime** Zeit in der **Inode** anzupassen. Das Mounten eines Dateisystems mit der `noatime` Option unterlässt dies und beschleunigt den Dateizugriff beim Lesen. Vergleiche `mount(8)`.

4.4.3 Links

Es gibt zwei Methoden, um eine Datei *foo* mit einem anderen Dateinamen *bar* zu verknüpfen.

- Ein **harter Link** (hard link) ist ein anderer Name für eine existierende Datei (`ln foo bar`),
- Ein **symbolischer Link** (symlink) verweist auf eine andere Datei über den Namen (`ln -s foo bar`).

Man vergleiche das folgende Beispiel für Änderungen der Anzahl der Links und die feinen Unterschiede beim Anwenden des `rm` Kommandos.

```
$ echo "ursprünglicher Dateiinhalt" > foo
$ ls -l foo
-rw-r--r--    1 osamu    osamu           4 Feb  9 22:26 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -l foo bar baz
-rw-r--r--    2 osamu    osamu           4 Feb  9 22:26 bar
lrwxrwxrwx    1 osamu    osamu           3 Feb  9 22:28 baz -> foo
-rw-r--r--    2 osamu    osamu           4 Feb  9 22:26 foo
$ rm foo
$ echo "neuer Dateiinhalt" > foo
$ cat bar
ursprünglicher Dateiinhalt
$ cat baz
neuer Dateiinhalt
```

Alle symbolischen Links haben die Dateizugriffsrechte “rwxrwxrwx” wie im obigen Beispiel gezeigt und deren Zugriffsrechte werden durch die Datei auf die sie zeigen diktiert.

Das Verzeichnis `.` verweist auf das Verzeichnis in dem es sich befindet, die Anzahl der Links für ein neues Verzeichnis beginnt bei 2. Das Verzeichnis `..` verweist auf das übergeordnete Verzeichnis, die Anzahl der Links eines Verzeichnisses erhöhen sich mit dem Erzeugen neuer Unterverzeichnisse.

4.5 Weiterführende Informationen

Es gibt viele gute Referenzen zum Einstieg in Unix. O'Reilly's Bücher sind gewöhnlich gute Anleitungen für alle Computerthemen. Das LDP Dokument Linux Tipps HOWTO (`http://www.tldp.org/HOWTO/Tips-HOWTO.html`) ist eine andere nützliche Informationsquelle. Man vergleiche 'Unterstützung für Debian' auf Seite 199 für weitere Ressourcen.

Kapitel 5

Aktualisieren einer Distribution auf `testing`

Offizielle Hinweise für die Aktualisierung finden sich unter `http://www.debian.org/releases/stable/releasenotes` und `http://www.debian.org/releases/testing/releasenotes` (befindet sich in Arbeit).

Der Prozess zur Aktualisierung des Systems auf `testing/unstable` ist:

- Zunächst ist das APT System auf die Woody Version zu aktualisieren, wenn das aktuelle System Potato ist, da Potato's APT nicht über die in Woody's `apt_preferences(5)` beschriebenen Fähigkeiten verfügt.
- Modifizierung der `/etc/apt/sources.list` und `/etc/apt/preferences` Dateien, um Verweise auf den "testing" Abschnitt der benutzten Repositories hinzuzufügen. Sollte es gewünscht sein, können auch Verweise auf den "unstable" Abschnitt hinzugefügt werden.
- Aktualisierung der Paketlisten und Installation aller Pakete die sich nun upgraden lassen.

5.1 Übergang von APT auf die Woody Version

Ein Netzwerk Upgrade des APT Systems und einiger Kernpakete zur Woody Version kann wie folgt nach Einbinden von `stable` in `/etc/apt/sources.list` erreicht werden, wenn Sie noch Potato nutzen.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

5.2 Vorbereitung des Übergangs (stable nach testing)

Das Verwenden der testing Ausgabe von Debian hat den Nebeneffekt, das Sicherheitsupdates nur langsam eingespielt werden. Seien Sie gewarnt.

Eine Aktualisierung auf testing über das Netzwerk kann folgendermaßen durchgeführt werden (starten Sie das Skript go-woody (examples/), um dies in einem Schritt zu erledigen):

Neuerstellung der existierenden sources.list Datei:

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
```

Erzeugen einer sauberen Liste von Repositories für stable:

```
# cd /
# apt-setup noprobe
  ... auswählen der Repositories, auf die mit HTTP oder FTP
      Methoden zugegriffen wird
```

Hinzufügen des testing Abschnitts zu dieser neuen Liste. Die deb-src Zeilen werden auskommentiert.

```
# cd /etc/apt
# grep -e "^deb " sources.list >sources.deb
# grep -e "^deb-" sources.list >sources.src
# sed -e "s/stable/testing/" sources.deb >>sources.list
# sed -e "s/stable/testing/" sources.src | \
     sed -e "s/^deb-/#deb-/" >>sources.list

# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
 Package: *
 Pin: release a=testing
 Pin-Priority: 600

 Package: *
 Pin: release a=unstable
 Pin-Priority: 50

 EOF
```

Optional kann “unstable” hinzugefügt werden.

```
# sed -e "s/stable/unstable/" sources.deb >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
     sed -e "s/^deb-/#deb-/" >>sources.list
```

Vergleiche ‘Grundlagen der Debian-Paketverwaltung’ auf Seite 66 für genaueres zum Tuning von /etc/apt/sources.list und /etc/apt/preferences.

Nun kann ein Update und Upgrade wie im nächsten Abschnitt beschrieben durchgeführt werden.

5.3 Aktualisierung des Debian Systems

Nach den korrekten Anpassungen von /etc/apt/sources.list und /etc/apt/preferences, kann eine Aktualisierung des Systems auf testing durchgeführt werden. Vergleichen Sie mit ‘Debian Paketverwaltung’ auf Seite 65 für die Grundlagen und sollten dabei Fehler auftreten, schauen Sie in ‘APT Upgrade Fehlersuche’ auf Seite 72.

5.3.1 Beste Upgrade Vorgehensweise mittels dselect

Wenn ein System zahlreiche Pakete enthält, die -dev-Pakete, etc. nutzen, wird die im folgenden beschriebene Vorgehensweise mittels dselect empfohlen, da hier eine feinere Kontrolle der Pakete möglich ist.

```
# dselect update # muss vor jeder Aktualisierung ausgeführt werden
# dselect select # Auswahl zusätzlicher Pakete
```

All Ihre aktuellen Pakete werden ausgewählt, wenn dselect startet. dselect verlangt dabei eventuell nach zusätzlichen Paketen, basierend auf Abhängigkeiten, Vorschlägen und Empfehlungen. Möchten Sie keine Pakete hinzufügen, drücken Sie einfach Q, um dselect erneut zu beenden.

```
# dselect install
```

Sie werden einige Paket Konfigurationsfragen während dieses Teils des Prozesses beantworten müssen, halten Sie also Ihre Notizen bereit und planen Sie genug Zeit dafür ein. Vergleiche ‘dselect’ auf Seite 67.

Nutzen Sie dselect. **Dies funktioniert immer :)**

5.3.2 Unerwünschte Vorgehensweise mittels `apt-get`

Von der unten beschriebenen Verwendung von `apt-get` für die Aktualisierung des Systems wird abgeraten, auch wenn dies weit verbreitet ist. Wenn ein System ab Woody oder höher ohne `dselect` aktualisiert werden muss, sollte `aptitude` und andere Alternativen in Erwägung gezogen werden.

Wenn auf einem System nur wenige Pakete installiert sind oder im Debian-Archiv keine größeren Änderungen vorhanden sind, kann die im folgenden beschriebene Vorgehensweise (manchmal) ausreichen.

```
# apt-get update  # muss vor jedem Upgrade ausgeführt werden
 ... aktualisieren des Systems einschließlich der Abhängigkeiten:
# apt-get upgrade # muss vor jedem Upgrade ausgeführt werden
 ... aktualisieren des gesamten Systems einschließlich der
     Abhängigkeiten:
# apt-get -u dist-upgrade
 ... aktualisieren und Beibehalten der aktuellen Einstellungen von
     dselect (neu, besser):
# apt-get -u dselect-upgrade   # verwenden des dselect Auswahl
```

Da diese Upgrade Methode `apt-get` nutzt, ist die Verwendung der *Empfehlungen* (“recommends”) und *Vorschläge* (“suggests”) nur eingeschränkt möglich. Siehe ‘Paketabhängigkeiten’ auf Seite 16.

Kapitel 6

Debian Paketverwaltung

Um die Netzlast in den Debian-Archiven zu reduzieren, sollten Sie einen lokalen HTTP Proxy mit `squid` für das Herunterladen von Paketen mittels APT einrichten. Wenn nötig muss die `http_proxy` Umgebungsvariable oder der `http` Wert in `/etc/apt/apt.conf` gesetzt werden. Dies erhöht die Leistung im Netzwerk enorm, insbesondere dann, wenn viele Debian-Rechner in einem LAN vorhanden sind.

Obwohl die Pinning-Eigenschaft von `apt_preferences(5)` sehr leistungsfähig ist, löst sie nicht alle Abhängigkeitsprobleme, da die Voraussetzungen für Abhängigkeiten an neueren Versionen anderer fundamentaler Programmpakete zerren.

Die Verwendung der in '`chroot`' auf Seite 114 beschriebenen Methode ist geeignet, um gleichzeitig die Stabilität beider Systeme zu sichern und auf die neuesten Versionen von Software zuzugreifen.

Dieses Kapitel bezieht sich auf ein Woody System, jedoch sollten die Informationen hier auch auf ein Potato System (mit Ausnahme von `apt_preferences(5)` und Themen die Bezug auf die `/etc/apt/preferences` Datei nehmen) zutreffen.

6.1 Einführung

Wenn das Lesen der Entwicklerdokumentation zu viel für Sie ist, dann lesen Sie zuerst dieses Kapitel um dann die ganze Vielfalt von Debian (`testing/unstable`) zu nutzen :-)

6.1.1 Grundlegende Werkzeuge

```
dselect   - menügesteuertes Paketverwaltungs-Werkzeug (top level)
dpkg      - installiert Pakete (Paketdatei orientiert)
apt-get   - installiert Pakete (Paketarchiv orientiert, CLI APT)
tasksel   - installiert Aufgaben-Profile (Verbund von Paketen)
aptitude  - installiert Pakete (Pakete & Profile, ncurses APT)
```

```
deity      - alternatives ncurses APT
synaptic, gsynaptic - GUI APT Alternativen
```

Diese Tools arbeiten nicht alle auf dem gleichen Level. `dselect` arbeitet ein Level über APT (der Kommandozeilenbefehl ist `apt-get`) und `dpkg`.

APT nutzt `/var/lib/apt/lists/*` um verfügbare Pakete zu erfassen, `dpkg` jedoch nutzt `/var/lib/dpkg/available`. Wenn Sie Pakete direkt mit `apt-get` oder mit einem ähnlichen Programm, wie z.B. `aptitude` installiert haben, dann stellen Sie sicher, dass Sie die Datei `/var/lib/dpkg/available` mittels `[U]pdate` aus dem `dselect` Menü oder von der Kommandozeile mit "`dselect update`" updaten, bevor Sie `dselect select`, `tasksel` oder `dpkg -l` ausführen.

Was die Paketabhängigkeiten angeht, so berücksichtigt `apt-get` automatisch alle **benötigten** Pakete, ignoriert aber **empfohlende** und **vorgeschlagene** Pakete. `dselect` bietet jedoch eine menübasierte Kontrolle über die Auswahl von solchen Paketen und fragt zur Auswahl von **benötigten**, **empfohlenden** und **vorgeschlagenen** Paketen. `aptitude` bietet die Möglichkeit, alle **benötigten**, **empfohlenden** und **vorgeschlagenen** Pakete automatisch zu berücksichtigen. Siehe 'Paketabhängigkeiten' auf Seite 16.

6.1.2 Nützliche Tools

```
apt-cache         - durchsucht das Paketarchiv im lokalen Cache
dpkg-reconfigure  - erneutes Konfigurieren von installierten
                    Paketen (sofern sie debconf benutzen)
dpkg-source       - verwaltet Quellpaketdateien
dpkg-buildpackage - automatisiert den Prozess der Erstellung von
                    Paketdateien
...
```

6.2 Grundlagen der Debian-Paketverwaltung

Man kann eine Menge von Paketen genannt *Aufgaben-Profile* oder auch *Task* installieren oder individuelle Pakete. Eine Aktualisierung des Systems kann auch mittels Paketverwaltungswerkzeugen wie weiter unten beschrieben erfolgen. Es wird auch auf 'Debian System-Installationshinweise' auf Seite 29, 'Aktualisieren einer Distribution auf `testing`' auf Seite 61 und 'Rettungseditoren' auf Seite 161 verwiesen.

6.2.1 *Aufgaben-Profile* mit `tasksel` oder `aptitude` installieren

`tasksel` ist das **Debian Aufgaben-Profil Installationsprogramm**, das auch während der Debian Installation unter dem Punkt "`einfach`" angeboten wird.

Wenn man eine bestimmte Funktion benötigt, die verschiedene Pakete erfordert, so ist das der einfachste Weg dies zu erledigen. Stellen Sie sicher, dass Sie die Befehle wie folgt starten:

```
# dselect update
# tasksel
```

`aptitude` bietet auch Zugang zu *Aufgaben-Profilen*. Dies erlaubt nicht nur die Auswahl von *Aufgaben-Profilen*, sondern ermöglicht es auch, Pakete in diesem *Profil* selektiv durch ein Menü abzuwählen.

6.2.2 Einrichten des APT Systems

Um selektive Upgrades durchzuführen, während man die `testing` Distribution installiert hat, muss das APT System (>Woody) wie in 'Übergang von APT auf die Woody Version' auf Seite 61 beschrieben eingerichtet werden, damit die `apt_preferences(5)` Eigenschaften genutzt werden können.

Fügen Sie zuerst die Quellen für `stable`, `testing` und `unstable` in Ihre `/etc/apt/sources.list` Datei ein. Editieren Sie danach `/etc/apt/preferences`, um Pin-Priority korrekt einzurichten.

```
Package: *
Pin: release a=stable
Pin-Priority: 500

Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50
```

6.2.3 `dselect`

Wenn `dselect` startet, wählt es automatisch alle Pakete aus, die in den Kategorien "Required", "Important" und "Standard" enthalten sind. In der Potato Distribution waren etliche große Pakete wie z.B. teTeX und Emacs in diesen Kategorien enthalten und wurden am besten für die Erstinstallation manuell abgewählt (mittels '_'). In der Woody Distribution sind diese Pakete nun in der "Optional" Paketkategorie.

`dselect` hat eine einigermaßen gewöhnungsbedürftige Benutzerschnittstelle. Es gibt gleich vier doppeldeutige Befehle (Großbuchstaben sind gemeint!):

```
Taste  Aktion
Q      Beenden. Momentane Auswahl bestätigen und beenden.
       (Abhängigkeiten vernachlässigen)
```

```
R       Rückgängig! Es war nicht so gemeint.
D       Verdammt! Es ist uninteressant was dselect meint. Mach es!
U       Alles auf den vorgeschlagenen Status setzen.
```

Mit D und Q kann eine Auswahl die Konflikte enthält, auf eigenes Risiko gewählt werden. Hiermit sollte vorsichtig umgegangen werden. Zur Zeit ist `dselect` das ausgereifteste menübasierte Werkzeug, das fein gegliederte Kontrolle über **Empfehlungen** und **Vorschläge** bietet.

Fügen Sie eine Zeile in `/etc/dpkg/dselect.cfg` ein, die die Option "expert" enthält, um die Ausgabe unnötiger Meldungen zu reduzieren. Für langsame Rechner, ist es empfehlenswert `dselect` auf einem anderen, schnelleren Rechner zu starten um Pakete zu finden und diese mit `apt-get install` zu installieren.

`dselect` erlaubt keinen Zugriff auf Pakete, die nicht durch Pin-Priority begünstigt sind.

6.2.4 `aptitude`

`aptitude` ist ein menübasiertes Paketinstallationprogramm ähnlich zu `dselect`. Dies kann auch als alternativer Kommandozeilen-Befehl an Stelle von `apt-get` genutzt werden. Vergleichen Sie `aptitude(1)`.

`aptitude` akzeptiert folgende Tastenkürzel, die meist Kleinbuchstaben sind.

```
Tastenkürzel  Aktion
F10           Menü
?             Hilfe zu Tastenkürzel
u             Aktualisiere Paketarchivinformation
g             Herunterladen und installieren gewählter Pakete
q             Beenden und Sichern der Änderungen
x             Beenden und Verwerfen der Änderungen
Enter         Informationen zu einem Paket anzeigen
```

`aptitude` erlaubt es alle Pakete, je nachdem, ob sie **benötigt**, **empfohlen** oder **vorgeschlagen** werden, auszuwählen. Dieses Verhalten kann durch Auswahl von `F10 -> Optionen -> Abhängigkeitsbehandlung` im Menü geändert werden.

`aptitude` erlaubt den Zugriff auf alle Versionen eines Pakets.

6.2.5 Die `apt-cache` und `apt-get` Kommandos

Beim Einrichten von `testing` wie im obigen Beispiel beschrieben, kann das System durch folgende Kommandos angepasst werden:

- `apt-get -u upgrade`

  Dies verwendet die `testing` Distribution und aktualisiert alle Pakete im System, wobei die durch Abhängigkeiten benötigten Pakete von `testing` stammen.

- `apt-get -u dist-upgrade`

  Dies verwendet die `testing` Distribution und aktualisiert alle Pakete im System, wobei alle Abhängigkeiten aufgelöst werden und die dadurch benötigten Pakete von `testing` stammen.

- `apt-get -u dselect-upgrade`

  Dies verwendet die `testing` Distribution und aktualisiert alle Pakete im System entsprechend der Auswahl durch `dselect`.

- `apt-get -u install` *`Paket`*

  Dies installiert *Paket* und dessen Abhängigkeiten aus der `testing` Distribution.

- `apt-get -u install` *`Paket`*`/unstable`

  Dies installiert *Paket* aus der `unstable` Distribution, während dessen Abhängigkeiten aus der `testing` Distribution installiert werden.

- `apt-get -u install -t unstable` *`Paket`*

  Dies installiert *Paket* aus der `unstable` Distribution, während dessen Abhängigkeiten auch aus der `testing` Distribution mit einer Pin-Priority von 990 für `unstable` installiert werden.

- `apt-cache policy` *`foo bar ...`*

  Dies gibt den Status der Pakete *foo bar ...* aus.

- `apt-cache show` *`foo bar ...`* `| less`

  Dies gibt die Informationen zu den Paketen *foo bar ...* aus.

- `apt-get install` *`foo=2.2.4-1`*

  Dies installiert die spezielle Version *2.2.4-1* des Pakets *foo*.

- `apt-get -u install` *`foo bar-`*

  Dies installiert das Paket *foo* und entfernt das *bar* Paket.

- `apt-get remove` *`bar`*

  Dies entfernt das Paket *bar*, die angepasste Konfiguration wird jedoch beibehalten.

- `apt-get remove --purge` *`bar`*

  Dies entfernt das Paket *bar* zusammen mit allen Konfigurationsdateien.

Die Option `-u` in den obigen Beispielen sorgt dafür, dass `apt-get` eine Liste aller Pakete ausgibt, die aktualisiert werden können und fragt den Nutzer bevor es beginnt. Das folgende setzt die Option `-u` standardmäßig:

```
$ cat >> /etc/apt/apt.conf << .
// Immer zu aktualisierende Pakete anzeigen (-u)
APT::Get::Show-Upgraded "true";
.
```

Die Option `-s` kann zum Simulieren eines Upgrades verwendet werden, ohne dass eine Aktualisierung erfolgt.

6.2.6 Verwenden einer speziellen Version der Debian Distribution

Abhängig von der von Ihnen bevorzugten Debian-Version kann die Datei `/etc/apt/preferences` in 'Einrichten des APT Systems' auf Seite 67 entsprechend Ihren Wünchen angepasst werden:

```
für stable:            Pin-Priority von testing auf 50 setzen
für testing:           Einstellungen wie oben gezeigt beibehalten
für testing(unstable): Pin-Priority von unstable auf 500 setzen
für unstable(testing): Pin-Priority von unstable auf 800 setzen
```

Eine Richtlinie für die Wahl der Pin-Priority ist das Wechseln von oben nach unten in der obigen Tabelle, so wie die Zeit von unmittelbar nach einer Veröffentlichung bis zum Einfrieren für die nächste Version vergeht.

Achtung: Das Verwenden der `testing` Version von Debian hat den Nebeneffekt, dass Sicherheitsupdates nur langsam eingespielt werden.

Wenn verschiedende Versionen von Debian verwendet werden, so wie `testing` mit `stable` oder `unstable` mit `stable`, so werden höchstwahrscheinlich wichtige Pakete unabsichtlich von `testing` oder `unstable` genutzt, die fehlerhaft sein können. Sie wurden gewarnt.

Beispiele von `/etc/apt/preferences`, in welchen einige wichtige Pakete auf ausgereifte Versionen gesetzt werden, während die nicht ganz ausgereiften für nichtessentielle Pakete verwendet werden, sind im Beispielverzeichnis (`examples/`) als `preferences.testing` und `preferences.unstable` verfügbar. Auf der anderen Seite erzwingt `preferences.stable` einen Versionssprung aller Pakete zu `stable` (Downgrad).

6.2.7 Downgrad aller Pakete zu `stable`

Um alle Pakete auf die `stable` Distribution zurückzustufen, editieren Sie die Datei `/etc/apt/preferences` wie folgt:

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

und führen Sie "`apt-get upgrade`" aus. Dieser Befehl stuft aufgrund der Pin-Priority > 1000 sämtliche Pakete herunter. Aber seien Sie gewarnt, es könnte kleinere Abhängigkeitsprobleme geben.

6.2.8 Überblick über `/etc/apt/preferences`

In dem in 'Einrichten des APT Systems' auf Seite 67 aufgeführten Beispiel von `/etc/apt/preferences` bedeutet "Package: *" in der ersten Zeile, dass sich der Abschnitt auf alle Pakete bezieht. Man kann auch einen Paketnamen anstatt von "*" verwenden, um die Pin-Priority für das Paket festzulegen.

"Pin: release a=stable" in der zweiten Zeile bedeutet, dass `apt-get` die `Packages.gz` Datei aus dem Archiv nimmt, das die Zeile "Archive: stable" in seiner `Release` Datei enthält.

Die letzte Zeile "Pin-Priority: 500" setzt Pin-Priority auf 500.

Wenn es mehrere Pakete mit dem selben Namen gibt, so wird normalerweise das mit der höchsten Pin-Priority installiert.

Die Bedeutung der Pin-Priority ist:

- 1001 und höher: Zurückstufungsprioritäten.

  Ein Zurückstufen (Downgrade) des Pakets ist erlaubt für diesen Bereich der Pin-Priority.

- 100 bis 1000: Standardprioritäten.

  Ein Zurückstufen des Pakets ist nicht erlaubt. Einige wichtige Pin-Priority Werte:

  - 990: die Priorität, die durch die `--target-release` oder `-t` Option von `apt-get(8)` gesetzt wird.
  - 500: die Priorität für alle Standardpaketdateien.
  - 100: die Priorität der zur Zeit installierten Paketdateien.

- 0 bis 99: Nicht-automatische Prioritäten. (Diese werden nur genutzt, wenn das Paket nicht installiert ist und keine andere Version verfügbar ist.)

- kleiner als 0: Die Version wird niemals gewählt.

Die selben Effekte wie `--target-release` können durch Setzen von `/etc/apt/apt.conf` mittels

```
# echo 'APT::Default-Release "testing";' >> /etc/apt/apt.conf
```

erreicht werden. Die Kommandozeilenoption `--target-release` und der Wert in `/etc/apt/apt.conf` ersetzen die Einstellung in `/etc/apt/preferences`. Seien Sie vorsichtig damit, wenn Sie gleichzeitig `/etc/apt/preferences` editieren.

6.3 Debian Überlebensbefehle

Mit diesem Wissen kann jeder ein Leben des **ewigen** "Aktualisierens" führen :-)

6.3.1 Überprüfen von Bugs in Debian und Suche nach Hilfe

Wenn Sie auf Probleme bezüglich eines bestimmten Pakets stoßen, stellen Sie sicher, dass Sie zuerst diese Seiten überprüfen, bevor Sie um Hilfe fragen oder einen Bug Report erstellen. (`lynx`, `links` und `w3m` sind hier gleichwertig):

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/Paketname # wenn der Name bekannt ist
$ lynx http://bugs.debian.org/Bugnummer # wenn Bugnr. bekannt ist
```

Suchen Sie auch bei Google (`www.google.de`) mit Suchbegriffen wie “site:debian.org”.

Wenn Sie nicht weiter kommen, lesen Sie die guten Handbücher. Setzen Sie `CDPATH` wie folgt:

```
export CDPATH=.:/usr/local:/usr/share/doc
```

und dann

```
$ cd Paketname
$ pager README.Debian # falls dies existiert
$ mc
```

Weitere Hilfequellen sind in ‘Unterstützung für Debian’ auf Seite 199 aufgelistet.

6.3.2 APT Upgrade Fehlersuche

Es können Paketabhängigkeitprobleme beim Aktualisieren auf `unstable/testing` entsprechend ‘Aktualisierung des Debian Systems’ auf Seite 63 entstehen. Meistens liegt das daran, dass ein Paket aktualisiert wird, das neue Abhängigkeiten hat, die nicht erfüllt sind. Diese Probleme können unter Verwendung folgender Befehle gelöst werden:

```
# apt-get dist-upgrade
```

Wenn dies nicht hilft das Problem zu lösen, dann wiederholen Sie folgende Befehle solange, bis sich das Problem auflöst:

```
# apt-get upgrade -f       # selbst bei Fehlern mit Upgrade ...
... oder
# apt-get dist-upgrade -f  # ... bzw. Dist-Upgrade fortfahren
```

Einige sehr fehlerhafte Upgrade-Skripte könnten permanente Schwierigkeiten bereiten. In solch einer Situation ist es gewöhnlich besser, wenn man sich die `/var/lib/dpkg/info/Paketname.{post,pre}{inst,rm}` Skripte der betreffenden Pakete ansieht und anschließend

```
# dpkg --configure -a      # konfiguriert alle nicht vollständig
                           # installierten Pakete
```

ausführt.

Wenn eines der Skripte sich über eine fehlende Konfigurationsdatei beschwert, dann schauen Sie in `/etc/` nach der entsprechenden Datei. Wenn eine mit der Dateiendung `.new` (oder so ähnlich) existiert, entfernen Sie diese Dateiendung einfach (z.B. mit `mv`).

Paketabhängigkeitsprobleme können auftreten, wenn Pakete in `unstable/testing` installiert werden. Diese können auf unterschiedliche Art und Weise umgangen werden.

```
# apt-get install -f Paket # ignoriere falsche Abhängigkeiten
```

Eine Alternative dazu wäre, das `equivs` Paket zu verwenden. Siehe `/usr/share/doc/equivs/README.Debian` und 'Das `equivs` Paket' auf Seite 83.

6.3.3 Wiederherstellung mit `dpkg`

Sollte das System einmal wirklich schwer geschädigt worden sein, beispielsweise wenn `dselect` (APT) abstürzt, kann man es auch nur mit Hilfe von `dpkg` wiederherstellen:

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i *             # solange, bis keine Fehler mehr auftreten
```

Sollte einmal ein Paket fehlen, können Sie es von Mirror-Seiten (`http://www.debian.org/misc/README.mirrors`) herunterladen:

```
# mc    # verwenden Sie "FTP-Verbindung" auf Debian FTP Server
```

Neue Pakete auf den HTTP/FTP Servern befinden sich nicht mehr in dem klassischen `dist` Verzeichnis, sondern im neuen `pool` Verzeichnis. (Siehe 'Das `pool` Verzeichnis' auf Seite 9.)

Die Installation des Pakets erfolgt dann folgendermaßen:

```
# dpkg -i /var/cache/apt/archives/Paketdatei.deb
```

Sollte ein Problem mit einer Abhängigkeit auftreten, kann dieses wie folgt gelöst werden:

```
# dpkg --ignore-depends=Paket1 ... -i Paketdatei.deb
# dpkg --force-depends -i Paketdatei.deb
# dpkg --force-depends --purge Paket
# dpkg --force-confmiss -i Paketdatei.deb # installiert fehlende
                                          # Konfigurationsdateien
```

6.3.4 Wiederherstellung von Paketauswahldaten

Wenn `/var/lib/dpkg/status` aus irgendeinem Grund fehlerhaft ist, so verliert das Debian System die Paketauswahldaten und nimmt Schaden. Schauen Sie nach der alten `/var/lib/dpkg/status` Datei in `/var/lib/dpkg/status-old` oder `/var/backups/dpkg.status.*`.

Eine eigene Partition für `/var/backups/` zu erstellen, könnte eine gute Idee sein, insbesondere da sich dort wichtige Systemdaten befinden.

Wenn die alte `/var/lib/dpkg/status` Datei nicht verfügbar ist, kann man dennoch Informationen von Verzeichnissen in `/usr/share/doc/` wiederherstellen.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  egrep -v '^(debian|texmf)$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
... neuinstallieren des Systems, abwählen je nach Bedarf:
# dselect --expert
```

6.3.5 Systemrettung nach Löschen von `/var`

Da das `/var` Verzeichnis regelmäßig aktualisierte Daten wie Mails enthält, ist es für einen Ausfall anfällig. Das Auslagern auf eine unabhängige Partition reduziert die Risiken. Wenn ein Unglück geschieht, muss das `/var` Verzeichnis wiederhergestellt werden, um das Debian System zu retten.

Besorgen Sie sich den wichtigsten Inhalt des `/var` Verzeichnisses von einem kleinen noch funktionsfähigem Debian System, basierend auf der selben oder einer älteren Debian Version, z.B. `var.tar.gz(http://people.debian.org/~osamu/pub/)` und kopieren Sie dies in das Stammverzeichnis des defekten Systems. Führen Sie nun folgendes aus:

```
# cd /
# mv var var-alt      # wenn noch sinnvolle Daten vorhanden sind
# tar xvzf var.tar.gz # Woody's Minimaldatei verwenden
# aptitude            # oder dselect
```

Dies sollte zu einem funktionierenden System führen. Die Wiederherstellung der Paketauswahl kann beschleunigt werden, wenn man die in 'Wiederherstellung von Paketauswahldaten' auf dieser Seite beschriebene Technik anwendet. (FIXME: Dieses Vorgehen benötigt mehr Versuche zur Bestätigung.)

6.3.6 Installation eines Pakets in ein nicht bootfähiges System

Booten Sie Linux von einer der Debian Rettungs-Disketten/CD's oder von einer alternativen Partition in einem Multi-Boot System. Vergleichen Sie 'Booten des Systems' auf Seite 95. Mounten Sie nun die nicht bootfähige Partition nach `/target` und verwenden Sie die chroot Installationsmethode von `dpkg`.

```
# dpkg --root /target -i Paketdatei.deb
```

Anschließend sollten alles nötige konfigurieren und Probleme beheben.

Übrigens, wenn `lilo` alles ist, dass Sie davon abhält das System zu booten, können Sie auch ganz einfach von einer Debian Rettungsdiskette booten. An dem Boot-Prompt, vorausgesetzt Ihre root Partition befindet sich auf `/dev/hda12` und Sie möchten im Run-Level 3 starten, geben Sie folgendes ein:

```
boot: rescue root=/dev/hda12 3
```

Daraufhin startet ein nahezu vollständiges System, mit dem Kernel von der Bootdiskette. (Möglicherweise werden nicht alle Hardwarekomponenten erkannt oder Module nicht geladen, dies liegt aber an dem Kernel von der Bootdiskette.)

6.3.7 Der dpkg Befehl funktioniert nicht mehr – was nun?

Ein funktionsgestörtes `dpkg` kann es verständlicherweise unmöglich machen, beliebige `.deb` Dateien zu installieren. Ein Vorgehen wie das unten aufgeführte wird Ihnen dabei helfen, diese Situation wieder in den Griff zu bekommen. (Natürlich können Sie `links` durch Ihrem favorisierten Browser ersetzen.)

```
$ links http://http.de.debian.org/debian/pool/main/d/dpkg/
  ... herunterladen von dpkg_version_arch.deb
$ su
password: *****
$ ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Für die `i386` Architektur kann auch `http://packages.debian.org/dpkg` als URL genutzt werden.

6.4 Debian Nirvana Befehle

Das **Verständnis** der folgenden Befehle ist eine essentielle Voraussetzung, um zum einem eventuellen Upgrade Problemen gewachsen zu sein und zum anderen um das Debian **Nirvana** zu erreichen. :-)

6.4.1 Informationen über eine Datei

Sie möchten den Namen des Pakets in Erfahrung bringen, welchem eine bestimmte Datei angehört:

```
... suche nach Muster in den installierten Paketdateien:
$ dpkg {-S|--search} Muster
... finde das Dateinamen-Muster in den Debian-Archivdateien:
$ zgrep -e Muster /local/copy/of/debian/Contents-i386.gz
```

Sie können auch spezialisierte Paketkommandos verwenden:

```
... dlocate: inkompatibel zu slocate (sichere Version von locate):
# apt-get install dlocate
... schnellere Alternative zu dpkg -L and dpkg -S:
$ dlocate filename
... installiert Pakete automatisch auf Anfrage:
# apt-get install auto-apt
... erzeugt eine Datenbank für auto-apt:
# auto-apt update
... sucht nach einem Muster in allen Paketen, ob sie installiert
... sind oder nicht:
$ auto-apt search Muster
```

6.4.2 Informationen über ein Paket

Die Debian-Paketarchive können durchsucht und angezeigt werden. Stellen Sie jedoch vorher sicher, dass APT die richtigen Archive in `/etc/apt/sources.list` findet. Wenn Sie sehen wollen, wie sich Pakete in `testing/unstable` gegen die aktuell installierten verhalten, so verwenden Sie `apt-cache policy`. Dies ist sehr nett.

```
# apt-get check           # aktualisiert den Cache und schaut nach
                          # fehlerhaften Paketen
$ apt-cache search Muster # sucht Pakete anhand Paketbeschreibung
$ apt-cache policy Paket  # Paketprioritäten/Distributions-Infos
$ apt-cache show -a Paket # zeigt die Paketbeschreibung in allen
```

```
                          # Distributionen
$ apt-cache showsrc Paket # zeigt die Paketbeschreibung des
                          # Quellpakets
$ apt-cache showpkg Paket # zeigt Paketinformationen für Debugging
# dpkg --audit|-C         # sucht nach nicht vollständig
                          # installierten Paketen
$ dpkg {-s|--status} Paket ... # Beschreibung eines installierten
                          # Pakets
$ dpkg -l Paket ...       # Status des installierten Pakets
$ dpkg -L Paket ...       # zeigt die Dateien an, die durch das
                          # Paket installiert wurden
```

`apt-cache showsrc` ist in Woody noch nicht dokumentiert, funktioniert aber :)

Sie können ebenfalls anstelle der oben genannte Befehle Paketinformationen in folgende Dateien finden (verwenden Sie beispielsweise `mc` zum Suchen):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

Ein Vergleich folgender Dateien kann Aufschluss darüber geben, was genau bei der letzten Installation passiert ist.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

6.4.3 Nicht-Interaktive Installation mit APT

Um Pakete nicht-interaktiv zu installieren, fügen Sie folgende Zeile zu `/etc/apt/apt.conf` hinzu:

```
Dpkg::Options {"--force-confold";}
```

Dies ist äquivalent zu `apt-get -q -y Paket`. Da dies alle Fragen automatisch mit “Ja” beantwortet, kann dies zu Problemen führen. Nutzen Sie diese Option deshalb mit Vorsicht. Lesen Sie `apt.conf(5)` und `dpkg(1)`.

Sie können jedes einzelne Paket auch später wie in ‘Installierte Pakete erneut konfigurieren’ auf dieser Seite beschrieben konfigurieren.

6.4.4 Installierte Pakete erneut konfigurieren

Verwenden Sie folgende Befehle, um bereits installierte Pakete erneut zu konfigurieren.

```
# dpkg-reconfigure --priority=medium Paket [...]
# dpkg-reconfigure --all   # Neukonfiguration aller Pakete
# dpkg-reconfigure locales # generiere zusätzliche locales
# dpkg-reconfigure --p=low xserver-xfree86 # konfiguriere den
                                           # X-Server neu
```

Führen Sie dies für `debconf` aus, wenn Sie den Dialogmodus von `debconf` permanent ändern möchten.

Einige Programme sind mit speziellen Konfigurationsskripten ausgestattet.

```
apt-setup    - erstellt die Datei /etc/apt/sources.list
install-mbr  - installiert einen Master-Boot-Record-Manager
tzconfig     - setzt die lokale Zeitzone
gpmconfig    - konfiguriert den gpm Maus-Daemon
sambaconfig  - konfiguriert Samba in Potato (Woody benutzt debconf)
eximconfig   - konfiguriert Exim (MTA)
texconfig    - konfiguriert teTeX
apacheconfig - konfiguriert Apache (httpd)
cvsconfig    - konfiguriert CVS
sndconfig    - konfiguriert das Soundsystem
...
update-alternatives - definiert Standardbefehle, z.B. vi für vim
update-rc.d         - System-V init-Skript-Management
update-menus        - das Debian Menüsystem
...
```

6.4.5 Deinstallation von Paketen

Entfernen eines Pakets, die Konfiguration wird nicht gelöscht:

```
# apt-get remove Paket ...
# dpkg  --remove Paket ...
```

Entfernen eines Pakets und dessen Konfiguration:

```
# apt-get remove --purge Paket ...
# dpkg    --purge        Paket ...
```

6.4.6 “Halten” älterer Pakete

Zunächst ein Beispiel: Möchte man, dass die Pakete `libc6` und `libc6-dev` beim Ausführen von `dselect` und `apt-get -u upgrade` nicht durch neuere Versionen ersetzt werden, so kann dies folgendermaßen erreicht werden:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u install` *`Paket`* bleibt davon jedoch unbeeinflusst. Um diese Pakete andererseits bei einem automatischen Systemdowngrade mittels `apt-get -u upgrade` oder `apt-get -u dist-upgrade` zurück zu halten, ist die Datei `/etc/apt/preferences` wie folgt zu erweitern:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

An dieser Stelle kann der Eintrag "`Package:`" keine Paketnamen wie z.B. "`libc6*`" enthalten. Wenn Sie jedoch alle Pakete, die im Zusammenhang mit dem `glibc` Quellpaket stehen, auf dem selben Stand halten möchten, müssen Sie diese alle explizit auflisten.

Folgender Befehl zeigt alle Pakete an, die den Status "hold" haben:

```
dpkg --get-selections "*" | grep -e "hold$"
```

6.4.7 Gemischtes `stable`/`testing`/`unstable` System

`apt-show-versions` kann alle verfügbaren Paketversionen und deren Zugehörigkeit zu den einzelnen Distributionen anzeigen.

```
$ apt-show-versions | fgrep /testing | wc
... Anzahl der installierten Pakete aus testing
$ apt-show-versions -u
... Liste aller Pakete für die es aktuellere Versionen gibt
$ apt-get install ‘apt-show-versions -u -b | fgrep /unstable‘
... aktualisiert alle Paket aus unstable auf die aktuellste Version
```

6.4.8 Löschen von zwischengespeicherten Paketdateien

APT speichert alle heruntergeladenen Paketdateien in `/var/cache/apt/archives/`. Diese können folgendermaßen gelöscht werden:

```
# apt-get autoclean # löscht nur nicht benötigte Paketdateien
# apt-get clean     # löscht alle zwischengespeicherten Paketdateien
```

6.4.9 Speichern/Kopieren der Systemkonfiguration

Erstellen einer lokalen Kopie der aktuellen Paketauswahl:

```
$ dpkg --get-selections "*" >meineAuswahl   # oder \*
```

"*" veranlasst, dass auch Pakete mit einem Status von "purge" in *meineAuswahl* gespeichert werden.

Diese Datei kann auf einen anderen Computer kopiert werden, um dort die gleichen Pakete zu installieren:

```
# dselect update
# dpkg --set-selections <meineAuswahl
# apt-get -u dselect-upgrade    # oder dselect install
```

6.4.10 Portierung eines Pakets auf die `stable` Distribution

Um selektive Upgrades in der stabilen Distribution zu machen, ist es möglich, ein Quellpaket in dieser Umgebung zu kompilieren. Dies verhindert viele Paketaktualisierungen, die durch Abhängigkeiten verursacht würden. Zuerst müssen folgende Zeilen der `/etc/apt/sources.list` Datei hinzugefügt werden:

```
deb-src http://http.us.debian.org/debian testing \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US \
 main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
 main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US \
 main contrib non-free
```

Aus Formatierungsgründen sind die Zeilen hier in je zwei Zeilen umgebrochen. Natürlich muss aber jede mit "deb-src" beginnende Zeile für sich alleine in `sources.list` stehen.

Nun ist der Quellcode zu beziehen und ein lokales Paket zu erstellen:

```
$ apt-get update  # aktualisiere die Quellpaketsuchliste
$ apt-get source Paket
$ dpkg-source -x Paket.dsc
$ cd paket-version
  ... schauen Sie nach benötigten Paketen (Build-depends in der
  ... .dsc Datei) und installieren Sie diese ebenso wie fakeroot
```

```
$ dpkg-buildpackage -rfakeroot

  ... oder (keine Signatur erstellen)
  ... verwenden Sie später "debsign" wenn nötig
$ dpkg-buildpackage -rfakeroot -us -uc

  ... Anschließend die Installation des Pakets
$ su -c "dpkg -i Paketdatei.deb"
```

Gewöhnlich ist es nötig, einige Pakete mit der "-dev" Erweiterung zu installieren, um Paketabhängigkeiten auflösen zu können. `debsign` ist im Paket `devscripts` enthalten. `auto-apt` kann die Auflösung von Abhängigkeiten erleichtern. Mittels `fakeroot` lässt sich unnötige Benutzung des root Accounts vermeiden.

In der Woody Distribution lassen sich diese Abhängigkeitsprobleme vereinfachen. Als ein Beispiel kompilieren wir `pine`, das nur im Quellcode vorliegt.

```
# apt-get build-dep pine
# apt-get source -b pine
```

6.4.11 Lokale Paketarchive

Möchte man ein lokales Paketarchiv erzeugen, welches mit APT und `dselect` kompatibel ist, so muss die Datei `Packages` erstellt werden und Paketdateien müssen in einem bestimmten Verzeichnis erstellt werden.

Ein lokales `deb` Archiv (Repository) das ähnlich zum offiziellen Debian-Archiv ist, kann wie folgt erstellt werden:

```
# apt-get install dpkg-dev
# cd /usr/local
# install -d pool # die Paketdateien werden hier gespeichert
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # anpassen von priority und section
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
```

```
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

Alternativ kann auch folgende, schnellere aber unsaubere, Methode verwendet werden:

```
# apt-get install dpkg-dev
# mkdir /usr/local/debian
# mv /irgend/ein/Paket.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
# echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Auf ein solches Archiv kann auch von entfernten Computern zugegriffen werden, indem der Zugriff auf diese Verzeichnisse durch HTTP oder FTP ermöglicht wird. Die Datei `/etc/apt/sources.list` muss dementsprechend angepasst werden.

6.4.12 Installation und Konvertierung eines fremden Binärpakets

Mit Hilfe des `alien` Pakets lassen sich binäre Pakete, wie sie in vielen Distributionen wie Redhat `rpm`, Stampede `slp`, Slackware `tgz` und Solaris `pkg` verwendet werden, in das Debian `deb` Paketformat konvertieren. Wenn Sie ein Paket von einer anderen Distribution verwenden möchten, können Sie es mit `alien` in Ihr bevorzugtes Paketformat konvertieren und anschließend installieren. `alien` unterstützt ebenfalls LSB Pakete.

6.4.13 Verifizierung installierter Paketdateien

`debsums` verifiziert installierte Paketdateien anhand deren MD5 Prüfsummen. Jedoch gibt es auch Pakete die keine MD5 Prüfsummen enthalten. Eine mögliche temporäre Lösung für Systemadministratoren ist folgende:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

von Joerg Wendland <joergland@debian.org> (nicht getestet).

6.4.14 Optimierte `sources.list`

Um es kurz zu machen, raffinierte Anstrengungen um `sources.list` optimal zu erzeugen führte nicht zu einer signifikanten Verbesserungen für mich in den USA. Ich wähle manuell einen nahe gelegenen Server mittels `apt-setup`.

`apt-spy` erzeugt `sources.list` automatisch, basierend auf der Wartezeit und der Bandbreite. `netselect-apt` erzeugt eine komplexere `sources.list`, nutzt aber eine einfachere Methode zur Bestimmung des besten Spiegels (ping Zeitvergleich).

```
# apt-get install apt-spy
# cd /etc/apt; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

6.5 Weitere Debian Besonderheiten

6.5.1 Der `dpkg-divert` Befehl

Datei**umleitungen** ist eine Methode, um `dpkg` dazu zu bringen, Dateien nicht in ihre Standardverzeichnisse zu installieren, sondern in **umgeleitete** Verzeichnisse. **Umleitungen** können durch Debian Skripte dazu genutzt werden, Dateien zu verschieben, wenn sie einen Konflikt verursachen. Systemadministratoren können auch eine Umleitung verwenden, um die Konfigurationsdatei eines Pakets zu überschreiben, oder wann immer einige Dateien (welche nicht als **conffiles** markiert sind) von `dpkg` nicht überschrieben werden sollen, wenn eine neuere Version eines Pakets installiert wird, das solche Dateien enthält (siehe 'Bewahren der lokalen Konfiguration' auf Seite 14).

```
# dpkg-divert [--add]  Datei # erstellen der "Umleitung"
# dpkg-divert --remove Datei # entfernen der "Umleitung"
```

Für gewöhnlich ist es ratsam `dpkg-divert` nicht zu verwenden wenn es nicht absolut notwendig ist.

6.5.2 Das `equivs` Paket

Wenn Sie ein Quellpaket kompilieren ist es empfehlenswert, es zu einem echten lokalen Debianpaket (`*.deb`) zu machen. Benutzen Sie `equivs` als letzten Ausweg.

```
Package: equivs
Priority: extra
Section: admin
Description: Circumventing Debian package dependencies
 This is a dummy package which can be used to create Debian
 packages, which only contain dependency information.
```

6.5.3 Alternative Befehle

Möchte man dass der Befehl `vi` `vim` ausführt, benutzen Sie `update-alternatives`:

```
# update-alternatives --display vi
...
```

```
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Einstellungen im Debian Alternativsystem werden in `/etc/alternatives` als symbolische Links gespeichert.

Um Ihre bevorzugte X Window Umgebung einzustellen, führen sie `update-alternatives` für `/usr/bin/x-session-manager` und `/usr/bin/x-window-manager` aus. Für weitere Informationen, siehe 'Angepasste X-Sitzung' auf Seite 133.

`/bin/sh` ist ein symbolischer Link auf `/bin/bash` oder `/bin/dash`. Aus Kompatibilitätsgründen zu alten Bash verunreinigten Skripten ist es sicherer, `/bin/bash` zu verwenden. Um die POSIX Konformität einzuhalten sollte jedoch `/bin/dash` verwendet werden. Mit dem Upgrade auf den 2.4 Linux Kernel geht die Tendenz zugunsten zu `/bin/dash`.

6.5.4 System-V `init` und Runlevel

Der Standardrunlevel in dem gebootet wird lässt sich in `/etc/inittab` einstellen.

Im Gegensatz zu anderen Distributionen gibt Debian die Verwaltung der Runlevel komplett in die Hände des Systemadministrators. Das Management des System-V artigen `init` wird in Debian von `update-rc.d` Skripten durchgeführt.

Das Starten von `/etc/init/`*`name`* in den Runlevel 1, 2, 3 und das Stoppen in 4, 5 mit der sequenziellen Prioritätsnummer 20 (Normal) lässt sich folgendermaßen einrichten:

```
# update-rc.d name start 20 1 2 3 . stop 20 4 5 .
```

Das Löschen symbolischer Links während das Skript in `init.d` noch vorhanden ist, funktioniert folgendermaßen:

```
# update-rc.d -f name remove
```

Beim Editieren der Runlevel schummel ich. Ich editiere Einträge manuell mit dem `mv` Befehl in der Shell-Eingabeaufforderung von `mc` während ich Linkeinträge mit Alt-Enter kopiere. Ein Beispiel:

```
# mv S99xdm K99xdm # xdm deaktivieren (X Display Manager)
```

Selbst das Deaktivieren eines Daemons bewerkstellige ich, indem ich ein `exit 0` an den Anfang eines `init.d` Skriptes einfüge. Dies ist ein schneller Hack. Schließlich sind diese Dateien Konfigurationsdateien (`conffiles`).

6.5.5 Deaktivierung von Daemon Diensten

Die Debian Distribution nimmt die Systemsicherheit sehr ernst und erwartet, dass der Systemadministrator kompetent ist. Eine einfache Administration steht so oft nur an zweiter Stelle und viele Daemon-Dienste sind in der Standard Installation mit dem höchsten Sicherheitslevel vorkonfiguriert oder mit den wenigsten Diensten (oder keinen) ausgestattet.

Führen Sie `ps aux` aus oder überprüfen Sie die Inhalte von `/etc/init.d/*` und `/etc/inetd.conf`, wenn Sie irgenwelche Zweifel (zu Exim, DHCP, ...) haben. Überprüfen Sie auch `/etc/hosts.deny` wie in 'Zugriffskontrolle mit PAM und login' auf Seite 119. Der `pidof` Befehl ist ebenfalls hilfreich (siehe `pidof(8)`).

X11 erlaubt standardmäßig keine (entfernten) TCP/IP Verbindungen in neueren Debian Versionen. Schauen Sie in 'TCP/IP-Verbindungen mit X' auf Seite 136. X Weiterleitung mittels SSH ist ebenfalls deaktiviert. Vergleichen Sie mit 'X-Fernverbindungen mit `ssh`' auf Seite 136.

Kapitel 7

Der Linux Kernel unter Debian

Debian hat eine eigene Methode einen Kernel und dazugehörige Module zu kompilieren. Weitere Infos sind in 'Debian und der Kernel' auf Seite 25 zu finden.

7.1 Neukompilieren des Kernels

Die Tools `gcc`, `binutils` und `modutils` aus Debian `unstable` sind sehr hilfreich, wenn man einen aktuellen Linux Kernel kompilieren möchte. Siehe `/usr/share/doc/kernel-package/README.gz` um offizielle Informationen zu erhalten, besonders der untere Teil ist interessant.

Einen eigenen Kernel zu kompilieren ist eine nicht ganz einfache Sache, womit selbst begnadete Entwickler Schwierigkeiten haben könnten, gerade auch weil diese Thematik eine sich ständig weiter entwickelnde ist.

Manoj Srivastava schrieb: `--initrd` benötigt einen Debian-eigenen cramfs Patch.

Herbert Xu schrieb: Nein, das tut es nicht. Das einzige was nötig ist, um ein anderes Dateisystem als CRAMFS zu nutzen, ist, MKIMAGE in `/etc/mkinitrd/mkinitrd.conf` zu setzen.

Seien Sie vorsichtig und ziehen Sie `/usr/share/doc/kernel-package/README.gz` von Manjo und Kent stets zur Hilfe. Desweiteren stellen Sie sicher, dass Sie immer die aktuellste Version des unstable `kernel-package` Paketes haben, wenn Sie einen instabilen Kernel kompilieren möchten

Für einen Kernel, der nur für eine Maschine kompiliert ist, wird `initrd` nicht benötigt. Weil ich aber möchte, dass mein Kernel nahezu der gleiche ist, wie der, der von den kernel-image Paketen bereitgestellt wird, nutze ich es trotzdem. Wenn Sie `initrd` nutzen möchten, lesen Sie bitte `mkinitrd(8)` und `mkinitrd.conf(5)`. Siehe auch `http://bugs.debian.org/149236`.

7.1.1 Die Debian Standard Methode

Beachten Sie Fehler Meldungen für `kernel-package`, `gcc`, `binutils` und `modutils`. Benutzen Sie neue Versionen von diesen Programmen wenn nötig.

Einen angepassten Kernel unter Debian zu kompilieren, bedarf ein wenig Sorgfalt. Verwenden Sie die Option `--append_to_version` zusammen mit `make-kpkg`, um mehrere Kernel zu verwalten.

```
# apt-get install debhelper modutils kernel-package libncurses5-dev
# apt-get install kernel-source-2.4.18   # installiere die aktuellste Version
# apt-get install fakeroot
# vi /etc/kernel-pkg.conf                # fügen Sie meinen Namen und meine e
$ cd /usr/src                            # Arbeitsverzeichnis
$ tar --bzip2 -xvf kernel-source-2.4.18.tar.bz2
$ cd kernel-source-2.4.18                # wenn das Ihre Kernel Sourcen sind
$ cp /boot/config-2.4.18-386 .config     # verwenden Sie aktuelle .config als
$ make menuconfig                        # den Anforderungen entsprechend zus
$ make-kpkg clean                        # muss laut 'man make-kpkg' ausgefüh
$ fakeroot make-kpkg --append_to_version -486 --initrd \
       --revision=rev.01 kernel_image \
       modules_image # modules_image ist für pcmcia-cs* und weitere...
$ cd ..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # installiert den Kernel
```

`make-kpkg kernel_image` führt `make oldconfig` und `make dep` aus. Verwenden Sie `--initrd` nicht, wenn initrd nicht ausgewählt wurde.

Wenn Kernel-Module aus pcmcia-cs oder ohne pcmcia benötigt werden, sollte im `make menuconfig` unter "General setup—>" der Eintrag "PCMCIA/CardBus support —>" ausgewählt und die Konfiguration auf "< > PCMCIA/CardBus support" gesetzt werden, beispielsweise durch Deaktivieren der Box.

Auf einer SMP Maschine, setzen Sie CONCURRENCY_LEVEL der `kernel-pkg.conf(5)` entsprechend.

7.1.2 Klassische Methode

Laden Sie den "reinen" Quellcode von

- Linux: `http://www.kernel.org/`
- pcmcia-cs: `http://pcmcia-cs.sourceforge.net/`

herunter oder verwenden Sie einen äquivalenten Quellcode in Debian und führen Sie folgende Schritte aus:

```
# cd /usr/src
# tar xfvz linux-Version.tar.gz
```

```
# rm -rf linux
# ln -s linux-Version linux
# tar xfvz pcmcia-cs-Version.tar.gz
# ln -s pcmcia-cs-Version pcmcia
# cd linux
# make menuconfig
... den Anforderungen entsprechend konfigurieren ...
# make dep
# make bzImage
... Konfiguration von lilo / grub ...
... kopieren von /usr/src/linux/arch/i386/boot/bzImage nach /boot ...
... /sbin/lilo oder grub ausführen
# make modules; make modules_install
# cd ../pcmcia
# make config
# make all
# make install
... Hinzufügen von benötigten Modulen zu /etc/modules
# shutdown -r now
... nun den neuen Kernel booten ...
```

7.1.3 Kernelheader

Die meisten Programme brauchen die Kernelheader Dateien nicht, und einige Programme lassen sich sogar nicht erfolgreich kompilieren, wenn die Header Dateien genutzt werden. Statt dessen sollten die Programme beim Kompilieren gegen die Header Dateien gelinkt werden, gegen die auch **`glibc` gelinkt wurde**. Auf einem Debian System sind diese Kernelheader Dateien in `/usr/include/linux` und `/usr/include/asm` zu finden.

Entgegen einigen veralteten Dokumentationen, sollten keine symbolischen Links in `/usr/src/linux` von `/usr/include/linux` und `/usr/include/asm` erstellt werden.

Es sollte, wenn ein Programm bestimmte Kernelheader Dateien für Kernel-spezifische Programme benötigt, statt dessen das Makefile so angepasst werden, dass der Pfad der einzubindenden Header Dateien folgende Pfade enthält *`Verzeichnis-bestimmter-Kernelheader`*`/include/linux` und *`Verzeichnis-bestimmter-Kernelheader`*`/include/asm`.

7.2 Der modularisierte Kernel 2.4

Die neuen Debian 2.4 Kernel, welche von `kernel-image-2.4.`*`NN`* zur Verfügung gestellt werden, sind sehr stark modularisiert. Es muss sichergestellt sein, dass Module die verwendet werden sollen auch aktiviert sind, damit der Kernel wie gewünscht arbeitet.

Obwohl ich viele Beispiele für `/etc/modules` im folgenden Abschnitt als schnellen Fix habe, höre ich immer wieder, dass der richtige Weg, diese Modul abhängigen Probleme zu lösen, ist Aliase für die Geräte in Dateien unterhalb von `/etc/modutils/` anzulegen.

Siehe `Documentation/*.txt` im Linux Quellcode, um genauere Informationen zu erhalten.

7.2.1 PCMCIA

`/etc/modules` sollte folgendes enthalten, damit PCMCIA korrekt funktioniert:

```
# ISA PnP Treiber
isa-pnp
# Lowlevel PCMCIA Treiber
# yenta_socket # in meinem Fall jedoch nicht nötig
```

Für den Rest von PCMCIA sorgen die Skripte (aus dem `pcmcia-cs` Paket) `depmod` und `kmod`. `isa-pnp` brauchte ich auch, weil mein Laptop ein altes ISA-PCMCIA Modell ist. Neuere Laptops mit Cardbus/PCMCIA sollten dies nicht mehr brauchen.

Ein Zitat vom großzügigen Miquel van Smoorenburg `<miquels@cistron.nl>`:

"Ich habe einfach alles was mit PCMCIA zu tun hat, inklusive cardmgr, von meinem Laptop gelöscht und einen 2.4 Kernel mit CardBus Unterstützung installiert, plus das neue `hotplug` Paket aus Woody.

Solange man nur 32-bit Karten hat, wird das pcmcia Paket nicht benötigt; 2.4 stellt die nötigen Karten-Dienste von sich aus zur Verfügung. Und der Standard tulip Treiber sollte wunderbar mit der Dlink Karte arbeiten.

– Mike."

7.2.2 SCSI

[NICHT GETESTET] `/etc/modules` braucht folgendes damit SCSI funktioniert:

```
# SCSI core
scsi_mod
# SCSI generic driver
sg
# SCSI disk
sd_mod
# Alle anderen HW Module
...
```

`depmod` sollte auf die oben genannten Module achten.

7.2.3 Netzwerk Funktionalität

`/etc/modules` sollte für zusätzliche Netzwerk Funktionalität folgendes enthalten:

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (in Reihenfolge)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
#ipfwadm
```

Oben genanntes ist nicht optimiert. `depmod` sollte auf die oben genannten Module achten.

7.2.4 EXT3 Dateisystem (> 2.4.17)

Die Journaling Funktionalität für das EXT3 Dateisystem mit einem vorkompilierten Kernel-Image (> 2.4.17) von Debian zu aktivieren, setzt die folgenden Schritte voraus:

```
# cd /etc; mv fstab fstab.old
# sed 's/ext2/ext3,ext2/g' <fstab.old >fstab
# vi /etc/fstab
... setzen Sie den root Dateisystem Typ auf "auto" anstelle von "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
... installieren Sie den aktuellsten Kernel und konfigurieren Sie LILO
dementsprechend
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... tune2fs für alle EXT2 Dateisysteme wiederholen
# shutdown -r now
```

Von nun an ist EXT3 Journaling aktiviert. Für den Fall, dass der Kernel EXT3 für root Partitionen nicht unterstützt, erweist sich `ext3,ext2` als Dateisystem "Typ" in der `fstab` als sehr hilfreich, da so sicher ein "Fall-Back" gemacht werden kann.

Wenn Sie schon zuvor einen 2.4 Kernel installiert haben, und ihn nicht erneut installieren möchten, dann führen Sie die oben genannten Befehle bis zu dem `apt-get` Kommando aus, und dann folgende:

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp
/lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... tune2fs für alle EXT2 Dateisysteme wiederholen
# shutdown -r now
```

Von nun an ist EXT3 Journaling aktiviert.

Wenn `/etc/mkinitrd/modules` nicht gesetzt war als `mkinitrd` ausgeführt wurde, und Sie während der Boot-Phase zusätzliche Module hinzufügen möchten:

```
... um eine Shell (5 sec.) an dem initrd Prompt zu erhalten, RETURN drücken
# insmod jbd
```

```
# insmod ext3 # modprobe ext3 sollte alles nötige berücksichtigen
# insmod ext2
# ^D
... weiter geht's mit dem Boot-Vorgang
```

Die Meldung "cramfs: wrong magic" könnte in den System Meldungen beim Boot-Vorgang (`dmesg`) auftreten, diese wird jedoch als harmlos eingestuft. Dieses Problem wurde in Sarge (2002/10) gelöst. Siehe http://bugs.debian.org/135537 und das EXT3 File System mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html) oder `/usr/share/doc/HOWTO/en-txt/mini/extra/ext3-mini-HOWTO.gz` um weitere Informationen zu erhalten.

Es wurde berichtet, dass auf einigen Systemen einige Kernel lock-ups auftraten wenn EXT3 aktiviert war, aber was meinen 2.4.17 Kernel angeht, hatte ich keine Probleme.

7.2.5 Realtek RTL-8139 Unterstützung in Kernel 2.4

Aus irgendwelchen Gründen heißt das RTL-8139 Modul ab dem 2.4er Kernel 8139too und nicht mehr rtl8139. Wenn Sie also von einem 2.2er Kernel auf einen 2.4er upgraden, ändern Sie einfach die `/etc/modules` dementsprechend ab.

7.2.6 Parallel Port Unterstützung

Von `kernel-image-2.4.*` wird Parallel Port Unterstützung als Modul angeboten. Aktivieren Sie es wie folgt:

```
# modprobe lp
# echo lp >> /etc/modules
```

Siehe auch `Documentation/parport.txt` im Linux Quellcode.

7.3 Tunen des Kernels mittels des proc Dateisystems

Das Verhalten des Linux Kernels kann dynamisch durch das proc Dateisystem geändert werden.

Für grundlegende Informationen über das Ändern von Kernelparametern durch das `/proc` Dateisystem, lesen Sie im Linux Quellcodeverzeichnis `Documentation/sysctl/*`.

Einige Beispiele der Kernelparametermanipulation können in `/etc/init.d/networking` und 'Eigenartige Probleme beim Zugriff auf einige Webseiten' auf Seite 47 gefunden werden.

Sehen Sie in `sysctl.conf(5)` wie man den Kernel zur Bootzeit mittels des `/proc` Dateisystems konfigurieren kann. Dazu wird das Skript `/etc/init.d/procps.sh` genutzt, das in der Regel aus `/etc/rcS.d/S30procps.sh` gestartet wird.

7.3.1 Zu viele geöffnete Dateien

Der Linux Kernel könnte unter Umständen die Meldung "Too many open files" ("Zu viele geöffnete Dateien") ausgeben. Der Grund hierfür ist, dass der Standardwert (8096) für `file-max` sehr klein gewählt wurde. Um dieses Problem zu lösen, führen Sie folgende Schritte als root aus:

```
# echo "65536"  > /proc/sys/fs/file-max  # für 2.2 und 2.4 Kernel
# echo "131072" > /proc/sys/fs/inode-max # nur für 2.2 Kernel
```

Alternativ können Sie folgendes in `/etc/sysctl.conf` eintragen, um diese Änderungen permanent zu aktivieren:

```
file-max=65536   # für 2.2 und 2.4 Kernel
inode-max=131072 # nur für 2.2 Kernel
```

7.3.2 Disk Flush Intervalle

Sie können Disk Flush Intervalle durch das proc Dateisystem ändern. Das folgende kürzt das Intervall von standardmäßig fünf Sekunden auf eine Sekunde.

```
# echo "40 0 0 0 100 30000 60 0 0"  > /proc/sys/vm/bdflush
```

Dies kann die Ein-/Ausgabe-Performance ein klein wenig negativ beeinflussen. Aber dies sichert den Dateiinhalt mit Ausnahme der letzten Sekunde, was kürzer ist als die standardmäßigen fünf Sekunden. Dies ist selbst für die Journaling Dateisysteme wahr.

7.3.3 Trägheit alter Maschinen mit wenig Speicher

Für einige alte Systeme mit wenig Speicher kann es immer noch sinnvoll sein, over-commit von Speicher durch das proc Dateisystem zu aktivieren:

```
# echo 1 > /proc/sys/vm/overcommit_memory
```

Kapitel 8

Debian Tipps

8.1 Booten des Systems

Weitere Informationen über den Boot-Prompt (also die Eingabeaufforderung beim Systemstart) gibt es beim LDP BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html).

8.1.1 "Ich habe das root-Passwort vergessen!" (1)

Sobald man Zugriff auf die Tastatur eines Rechners hat, kann das System gestartet werden und ein Login stattfinden, auch ohne das root-Passwort zu kennen. (Dies setzt allerdings voraus, dass keine zusätzlichen Schutzmaßnahmen ergriffen wurden, wie z.B. ein BIOS-Passwort oder eine Passwortabfrage durch `lilo`, was ein Booten des Systems verhindern würde.)

Dies ist ein Vorgehen, das keine externen Boot-Medien und Änderungen der BIOS-Einstellungen benötigt. Die Bezeichnung für die voreingestellte Boot-Option sei "Linux".

Sobald die `lilo`-Boot-Meldung `boot:` erscheint, (auf einigen Systemen muss die Shift-Taste/Umschalt-Taste gedrückt werden, um einen automatischen Bootvorgang zu unterbrechen), ist folgendes einzugeben:

```
boot: Linux init=/bin/sh
```

Das System startet nun den Kernel und daraufhin das Programm `/bin/sh` anstelle des üblichen `init`-Prozesses. Die nun erlangte Shell hat root-Privilegien. Da das Hauptverzeichnis `/` nur les- aber nicht schreibbar ist und viele während des üblichen Boot-Prozesses eingeklinkten Verzeichnisse nicht erreichbar sind, müssen diese nachträglich gemountet werden, um ein vernünftiges Arbeiten zu ermöglichen:

```
init-2.03# mount -n -o remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
```

```
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

Wenn in der Datei `/etc/passwd` das zweite Datenfeld (der Trenner ist der Doppelpunkt) bei allen Einträgen ein "x" ist, dann benutzt das System shadow-Passwörter. In diesem Fall muss statt `/etc/passwd` die Datei `/etc/shadow` editiert werden. Um das root-Passwort zurückzusetzen, muss das zweite Feld des root-Eintrags gelöscht werden. Nach einem sauberen Neustart ist das root-Passwort leer und kann neu vergeben werden. Wenn das System in den Runlevel 1 fährt, wird (zumindest bei Debian-Versionen >= Potato) die Vergabe eines nichtleeren root-Passwortes eingefordert.

Für den gezeigten und ähnliche gelagerte Fälle ist ein Editor im `/bin`-Verzeichnis absolut empfohlen, für den Fall, dass der `/usr`-Baum nicht erreichbar ist (siehe dazu auch 'Rettungseditoren' auf Seite 161).

Für Notfälle hat sich außerdem die `sash` (stand alone shell) bewährt. Wenn das System nicht mehr gebootet werden kann, geben Sie am `lilo`-Prompt ein:

```
boot: Linux init=/bin/sash
```

Das Programm `sash` dient als Ersatz für die Shell `/bin/sh`. Es ist statisch gelinkt und hat viele Standardbefehle eingebaut. (Wenn `sash` gestartet ist, wird "help" eine kurze Referenz anzeigen.)

8.1.2 "Ich habe das root-Passwort vergessen!" (2)

Starten Sie das System mit einer Rettungsdiskette/-CD. Sei */dev/hda3* die root-Partition. Dann kann die Passwortdatei editiert werden durch:

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

Der Vorteil dieser Variante ist, dass ein eventuell vergebenes `lilo`-Passwort umgangen werden kann. Erforderlich ist allerdings, dass im BIOS die Einstellungen für das Boot-Laufwerk manipuliert werden kann (falls es nicht schon auf A: oder CDROM eingestellt ist).

8.1.3 Das System kann nicht gestartet werden

Dies ist kein Problem, selbst wenn Sie keine Boot-Diskette während der Installation angelegt haben. Wenn nur der `lilo` defekt ist, können Sie die Boot-Disk des Debian Installationsets verwenden und davon Ihr System booten. Wenn Ihre root-Partition */dev/hda12* ist und im Runlevel 3 gestartet werden soll, geben Sie folgendes am Boot-Prompt an:

```
boot: rescue root=/dev/hda12 3
```

Danach haben Sie ein fast vollständig funktionsfähiges System mit dem Kernel des Installationsmediums gestartet. (Es kann einige kleinere Probleme aufgrund fehlender Kerneleigenschaften oder Module geben.)

Vergleichen Sie 'Installation eines Pakets in ein nicht bootfähiges System' auf Seite 75, sollten Sie ein defektes System haben.

Benötigen Sei eine angepasste Boot-Diskette, folgen Sie den Anweisungen in `readme.txt` auf der Rettungsdiskette.

8.1.4 "Let me disable X on boot!"

Chasing `unstable/sid` is fun, but buggy `xdm`, `gdm`, `kdm`, and `wdm` started during the boot process can bite you bad.

First get the root shell by entering followings at the boot prompt:

```
boot: Linux vga=normal s
```

Here, *Linux* is the label for the kernel image you are booting "vga=normal" will make sure lilo runs in normal VGA screen, and "s" (or "S") is the parameter passed to `init` to invoke single user mode. Enter the root password at the prompt.

There are few ways to disable all the X starting daemons:

- run `update-rc.d ?dm stop 99 1 2 3 4 5 6`
- insert "exit 0" at the start of all `/etc/init.d/?dm` files.
- rename all `/etc/rc2.d/S99?dm` files to `/etc/rc2.d/K99?dm`.
- remove all `/etc/rc2.d/S99?dm` files.
- run `:>/etc/X11/default-display-manager`

Here, `rc2.d` must correspond to the runlevel specified in the `/etc/inittab`. Also `?dm` means all of the `xdm`, `gdm`, `kdm`, and `wdm`.

Only the first one in the list is "the one true way" in Debian. Last one is easy but only works on Debian and requires you to set it again later using `dpkg-reconfigure`. Others are generic methods to disable daemons.

You can still start X by `startx` command from any console shell.

8.1.5 Weitere Tricks mit dem Boot-Prompt

Mittels des `lilo`-Boot-Prompts kann das System in einen beliebigen Runlevel gestartet werden. Details dazu bietet: BootPrompt-HOWTO (`http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html`) (LDP).

Soll das System in den Runlevel 4 starten, dient dazu der Bezeichner des Standardsystem (s.o., "Linux") neben dem gewünschten Runlevel:

```
boot: Linux 4
```

Der Einzelbenutzer-Modus (nur für Wartungszwecke gedacht) wird mit einem der folgenden Kommandos erreicht:

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

Auch der benutzte Speicherbereich kann eingeschränkt werden. Das folgende Beispiel weist das System an, nur 48 MB des verfügbaren Speichers zu belegen.

```
boot: Linux mem=48M
```

Achtung: Wird an dieser Stelle mehr Speicher angegeben als vorhanden, wird der Kernel abstürzen. Wenn man mehr als 64 MB Speicher hat, kann es bei alten Kernels und/oder Mainboards passieren, dass das System nur 64 MB nutzt. In diesem Fall kann man versuchen den ungenutzten Speicher mit `mem=128M` zu aktivieren. Äquivalent dazu ist ein Eintrag in der Datei `/etc/lilo.conf`.

8.1.6 How do I set boot parameters (GRUB)

GRUB is a new boot manager from the GNU Hurd project and is much more flexible than Lilo but has slightly different handling of boot parameters.

```
grub> find /vmlinuz
grub> root (hd0,0)
grub> kernel /vmlinuz root=/dev/hda1
grub> initrd /initrd
grub> boot
```

Here, you must be aware of the Hurd device names:

```
Hurd/GRUB          Linux              MSDOS/Windows
 (fd0)              /dev/fd0           A:
 (hd0,1)            /dev/hda1          C: (usually)
 (hd0,4)            /dev/hda4          F: (usually)
 (hd1,4)            /dev/hdb4          ?
```

See `/usr/share/doc/grub/README.Debian` and `/usr/share/doc/grub-doc/html/` for the detail.

8.2 Aufzeichnung und Abspielen von Aktivitäten

8.2.1 Shell-Eingaben

Um ein Unix-artiges System gekonnt zu administrieren, sind mitunter knifflige oder raffinierte Aufgaben zu erledigen. Machen Sie sich deshalb mit den grundlegenden Konfigurationsaufgaben vertraut, so dass Sie im Notfall wissen, wo Sie Hand anlegen müssen. Fensterbasierte Konfigurations-Werkzeuge sind nett und bequem, nützen aber nichts, wenn man versucht eine defekte X-Window-Konfiguration wieder herzustellen.

Die Aufzeichnung von Tastatur-Eingaben ist, insbesondere als root, hin und wieder nützlich.

Emacs: `M-x shell` startet die Aufzeichnung. (“M” steht hier für die Meta-Taste, meist Alt oder auch Esc.) `C-x C-w` schreibt die aufgezeichnete Sequenz in eine Datei.

Shell: Das `script`-Kommando dient auf Shell-Ebene zum Aufzeichnen. with “^A H” as described in ‘Konsolenumschaltung mit `screen`’ auf Seite 111 or `script` command.

```
$ script
Script started, file is typescript
 ... irgendwelche Eingaben ...
 Ctrl-D
$ col -bx <typescript >savefile
$ vi savefile
```

Falls das `script`-Programm fehlt, kann diese Funktion mit der Shell simuliert werden:

```
$ bash -i 2>&1 | tee typescript
```

8.2.2 X Aktivitäten aufzeichnen

Um den grafischen Bildschirm aufzunehmen, auch X-Terminals und andere beliebige Fenster, kann `gimp` benutzt werden. Es besteht meist die Möglichkeit nur ein aktives Fenster oder den ganzen Schirm aufzunehmen und als Bilddatei auf der Festplatte zu speichern. Alternativen sind `xwd` (`xbase-clients`), `import` (`imagemagick`), oder `scrot` (`scrot`).

8.3 Kopieren und Archivieren eines Verzeichnisbaumes

8.3.1 Grundlegende Kommandos zum Kopieren eines Unterverzeichnisses

Möchte man seine Verzeichnisstruktur neu arrangieren, kopiert man den Inhalt eines Verzeichnisses `directory` von `/source` nach `/destination` durch die Standardmethode (soft links bleiben erhalten):

```
# cp -a /source/directory /dest/directory # erfordert GNU cp
# (cd /source/directory && tar cf - . ) | \
        (cd /dest/directory && tar xvfp - )
```

Wenn hard links im Unterverzeichnis vorliegen, ist etwas mehr Aufwand nötig:

```
# cd /path/to/old/directory
# find . -depth -print0 | afio -p -xv -0a /ort/neues/verzeichnis
```

Kopieren über eine Netzwerkverbindung:

```
# (cd /source/directory && tar cf - . ) | \
        ssh user@host.dom (cd /dest/directory && tar xvfp - )
```

Netzwerkkopie, falls keine Links vorliegen:

```
# scp -pr user1@host1.dom:/source/directory \
          user2@host2.dom:/dest/directory
```

(`scp` <==> `rcp` und `ssh` <==> `rsh`.)

Der folgende Vergleich über die verschiedenen Kopiermodi stammt von Manoj Srivastava <srivasta@debian.org> präsentiert in der Debian-Mailing-Liste debian-user@lists.debian.org.

8.3.2 `cp`

Das ursprüngliche `cp`-Programm war für nur wenige Zwecke geeignet. Es konnte keine Links auflösen und war auch für zerstreute Dateien wenig brauchbar.

Die GNU-Version von `cp` überwand zwar diese Probleme, doch auf nicht-GNU-Systemen kann es noch zu Schwierigkeiten kommen. Desweiteren kann `cp` keine portablen Archive erzeugen. Dennoch ein Beispiel:

```
% cp -a . neuesverzeichnis
```

8.3.3 `tar`

Das Archivierungsprogramm `tar` kann im Gegensatz zu `cp` mit symbolischen Links umgehen. Zusätzlich kann `cpio` allerdings auch mit "special files" umgehen.

Wenn `tar` auf mehrfache hard links einer Datei stößt, wird doch nur einmal die Datei in das Archiv kopiert; die Datei kann dann *nur* unter dem Namen der ursprünglichen Datei zurückgewonnen werden. `cpio` dagegen kopiert für jeden hard link die komplette Datei in das Archiv, so dass sie unter jedem der Link-Namen wiedergeholt werden kann.

Die Autoren des `tar`-Kommandos haben die Option für das Packen mit `bzip2` umbenannt. Empfohlen wird in Skripts die lange Option `--bzip2` zu benutzen. Die Kurzform `-I` (Potato) oder `-j` (Woody) könnte zu Portabilitätsproblemen führen.

8.3.4 `pax`

Hinter dem Namen `pax` (POSIX-Standard (IEEE Std 1003.2-1992, Seite 380–388 (section 4.48) und Seite 936–940 (section E.4.48)), verbirgt sich ein multifunktionales Werkzeug zum Austausch von Archiven (Portable Archive Interchange utility). `pax` liest und schreibt Archive und listet deren Inhalt, kopiert Verzeichnisse hinein und heraus und all dies unabhängig von einem spezifischen Archivformat. Mit anderen Worten, eine große Bandbreite von Formaten wird unterstützt.

`pax` ist noch neu und es können noch Kinderkrankheiten auftreten.

```
# apt-get install pax
$ pax -rw -p e . newdir
 oder
$ find . -depth  | pax -rw -p e  newdir
```

8.3.5 `cpio`

Das Kommando verwaltet die gleichnamigen Archive oder auch solche, die mit `tar` erstellt wurden. Das Archiv kann statt einer Datei auch eine Pipe oder ein Magnetband sein:

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

8.3.6 `afio`

`afio` ersetzt `cpio`. Es liest dessen Archive schneller, unterstützt mehr Laufwerke, hat eine bessere Fehlerkorrektur und kann das Archiv auf mehrere Medien verteilen. Die Kompression mit `afio` ist zuverlässiger als die von `tar` und `cpio`. Also benutzt man am besten `afio` statt derer. Zum Beispiel in einem Skript, das folgende Zeilen enthält:

```
$ find . -depth -print0 | afio -px -0a new-dir
```

Ich mache alle meine Backups auf Band mit `afio`.

8.4 Differential backup and data synchronization

Differential backup and data synchronization can be implimented with several methods:

- `rcs`: backup and history, text-only
- `rdiff-backup`: backup and history. symlink OK.
- `rsync`: 1-way synchronization
- `unison`: 2-way synchronization
- `cvs`: multi-way synchronization with server backup and history, text-only, mature. See 'CVS' auf Seite 167.
- `arch`: multi-way synchronization with server backup and history, no such thing as a "working directory".
- `subversion`: multi-way synchronization with server backup and history, Apache.

Combination of one of this with the archiving method described in 'Kopieren und Archivieren eines Verzeichnisbaumes' auf Seite 100 and the automated regular job described in 'Terminplanung für Prozesse (`cron`, `at`)' auf Seite 110 will make a nice backup system.

I will explain 2 easy-to-use utilities.

8.4.1 Differential backup with rdiff

`Rdiff-backup` offers nice and simple backup with the differential history for any types of files including symlinks. To back up most of `~/` to `/mnt/backup`:

```
$ rdiff-backup --include ~/tmp/keep --exclude ~/tmp  ~/ /mnt/backup
```

To restore 3 day old data from this archive to `~/old`:

```
$ rdiff-backup -r 3D /mnt/backup ~/old
```

See `rdiff-backup(1)`.

8.4.2 Regular differential backup with RCS

`Changetrack` will record changes to the text-based configuration files in RCS archives regularly. See `changetrack(1)`.

```
# apt-get install changetrack
# vi changetrack.conf
```

8.5 Einfrieren von Prozessen

8.5.1 Einen Prozess killen/beenden

`top` (table of processes) hilft außer Kontrolle geratene Prozesse zu identifizieren. 'P' sortiert die Spalten nach CPU-Last, 'M' nach Speicherverbrauch und 'k' kann einen Prozess "abschießen". Alternatively, BSD style `ps aux | less` or System V style `ps -efH | less` may be used. The System V style syntax displays parents process ID `PPID` which can be used for killing zombie (defunct) child.

Von der Kommandozeile hat `kill` die gleiche Funktion. Statt den Prozess nur zu beenden gibt es aber auch die Möglichkeit, fein abgestufte Signale an Prozesse zu senden. Während `kill` einen Prozess über die Prozess-Identifikationsnummer adressiert, ist es leichter den Prozess mittels `killall` über den Namen des Programms anzusprechen. Die meist gebrauchten Signale sind

```
 1: HUP,  Daemon neustarten
15: TERM, Normales Beenden
 9: KILL, Sofortiges Beenden!
```

8.5.2 ALT-SysRq

Wenn der Kernel mit der "magischen SysRq Taste" kompiliert wurde, kann das System mit etwas Glück auch aus dem totalen Nirvana geholt werden. Drücken der Tasten `ALT-SysRq` bei einem i386, gefolgt von einer der Tasten `r 0 k e i s u b`, aktiviert die Kernel-Notbremse.

'r' = restore, versucht die Tastatur wieder zu beleben, wenn X abstürzt. '0' setzt den Level mit dem Fehlermeldungen auf der Konsole ausgegeben werden herunter. 'k' = (system attention key) beendet alle Prozesse auf der aktuellen virtuellen Konsole. 't' beendet alle Prozesse des aktuellen Terminals außer init. 'i' beendet *alle* Prozesse außer init.

's'ync, 'u'mount und re'b'oot sind wirklich nur für den allerletzten Notfall.

Debian Standard (default) Kernels sind nicht mit dieser Option kompiliert (11/2002). Es muss zur Nutzung dieser Tasten einen neuer Kernel kompiliert werden. Weitere Informationen zu dieser "Notbremse" in `/usr/share/doc/kernel-doc-version/Documentation/sysrq.txt.gz` oder `/usr/src/`*`kernel-version`*`/Documentation/sysrq.txt.gz`.

8.6 Raffinierte merk-würdige Kommandos

8.6.1 Dateibetrachter

`less` ist ein Dateibetrachter, der Textdatei seitenweise ausgibt. Hilfe bietet 'h'. `less` kann mehr als dessen Urahn `more`. Mit `eval $(lesspipe)` können Filter eingerichtet werden, um so z.B. gepackte Dateien anzusehen. Diese Filter können auch permanent eingerichtet werden, indem man Filter in einer der Shell-Startdateien (mit `eval $(lessfile)`) festlegt. Siehe hierzu auch `/usr/share/doc/lessf/LESSOPEN`. Die Option `-R` erlaubt die Ausgabe von Sonderzeichen (raw characters) und schaltet die ANSI Farbsequenzen ein. Siehe `less(1)` für noch mehr Möglichkeiten.

`w3m` kann für einige Systeme eine Alternative zum Betrachten von Dateiinhalten sein. (EUC).???

8.6.2 Freier Speicher

`free` und `top` informieren über freien Speicher und dessen Verbrauch. Ausschlaggebend ist an dieser Stelle der "used"-Eintrag in der zweite Datenzeile (hier: 38792)

```
$ free -k # für 256MB Hauptspeicher
             total       used       free     shared    buffers cached
Mem:        257136     230456      26680      45736     116136 75528
-/+ buffers/cache:      38792     218344
Swap:       264996          0     264996
```

Die präzise Menge an Hauptspeicher, die zur Verfügung steht erfährt man außerdem mit `grep '^Memory' /var/log/dmesg`, was in diesem Fall "Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)" ergibt.

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

Ca. 5MB können nicht vom System verwendet werden, da der Kernel sie in Beschlag nimmt.

8.6.3 Setzen der Uhrzeit (BIOS)

```
# date MMDDhhmmCCYY
# hwclock --utc
# hwclock --systohc
# hwclock --show
```

Dies setzt die Hardware-Uhr auf die Zeit DD=Tag, MM=Monat, hh=Stunde, mm=Minute, CCYY=Jahr. Die Systemuhr wird dann mit der Hardwareuhr synchronisiert und zudem wird UTC (Coordinated Universal Time) als Zeitzone definiert.

8.6.4 Setzen der Uhrzeit (NTP)

Siehe: Managing Accurate Date and Time HOWTO (http://www.tldp.org/HOWTO/TimePrecision-HOWTO/index.html).

Setzen der Zeit bei permanenter Internetverbindung

Die Uhrzeit kann mit Hilfe eines Zeit-Servers gesetzt werden:

```
# ntpdate server
```

Durch ein Skript in `/etc/cron.daily` kann man so täglich die Rechnerzeit mit dem Zeit-Server synchronisieren.

Setzen der Uhrzeit bei nur sporadischer Internet-Verbindung

Das Paket `chrony` hilft hier weiter.

8.6.5 Den Bildschirmschoner deaktivieren

Auf der Konsole

```
# setterm -powersave off
```

Start der kon2(kanji) Konsole mit:

```
# kon -SaveTime 0
```

Während X-Windows läuft

```
# xset s off
 oder
# xset -dpms
 oder
# xscreensaver-command -prefs
```

Weiteres liefern die entsprechenden `man`-Seiten.

8.6.6 Durchsuchen von Datenbanken zur Systemverwaltung

Die glibc-Bibliothek ermöglicht das Durchsuchen von System-Datenbanken, z.B. passwd, group, hosts, services, protocols, networks. Der Befehl dazu lautet `getent`

```
getent database [key ...]
```

8.6.7 Sound abstellen (beep)

Im Zweifelsfall kann man immer den Stecker des Lautsprechers herausziehen ;-) Für die Bash-Shell gilt:

```
echo "set bell-style none">> ~/.inputrc
```

8.6.8 Fehlermeldungen auf der Konsole

Wenn übermäßig viele Fehlermeldungen die Konsole unbrauchbar werden lassen, sollte man zuerst in `/etc/init.d/klogd` nachsehen. Um den Warnlevel zu ändern kann man hier `KLOGD="-c 3"` setzen. Neustart des Daemons mit `/etc/init.d/klogdrestart` aktiviert die neuen Schwellen. Alternativ kann `dmesg -n3` benutzt werden. Der Warnlevel schlüsselt sich wie folgt auf:

- 0: KERN_EMERG, System ist unbenutzbar
- 1: KERN_ALERT, Sofortiger Eingriff nötig
- 2: KERN_CRIT, Kritischer Zustand
- 3: KERN_ERR, Fehler
- 4: KERN_WARNING, Warnung
- 5: KERN_NOTICE, Notiz
- 6: KERN_INFO, Information
- 7: KERN_DEBUG, Reine debug Nachricht

Wenn eine bestimmte, mit Sicherheit unkritische Fehlermeldung oft auftritt, kann diese auch durch einen trivialen Kernelpatch unterbunden werden. (Siehe dazu das Beispiel `shutup-abit-bp6` in examples subdirectory (`examples/`))

Auch ein Blick in `/etc/syslog.conf` kann hilfreich sein, um zu verstehen, welche Nachrichten auf der Konsole ausgegeben werden.

8.6.9 Setzen des korrekten Konsole-Typs

Die Konsole wird in Unix-artigen Systemen üblicherweise mit den Routinen aus der (n)curses-Bibliothek angesteuert. Dies erlaubt eine im wesentlichen von der Terminal-Art unabhängige Ausgabe mit vernünftiger Update-Strategie. Siehe `ncurses(3X)` und `terminfo(5)`.

Debian unterstützt eine ganze Reihe von Voreinstellungen für die Konsole.

```
$ toe | less                  # alle Einträge
$ toe /etc/terminfo/ | less   # Benutzer einstellbar
```

Die Auswahl kann durch Export der Umgebungsvariablen `TERM` aktiviert werden.

Wenn der terminfo-Eintrag für xterm mit einem xterm, das remote aufgerufen wird und das nicht auf Debian läuft, nicht funktioniert, kann der Terminaltyp von "xterm" auch auch eine der primitiveren Varianten, wie "xterm-r6", umgestellt werden. Siehe `/usr/share/doc/libncurses5/FAQ` für weitere Informationen Der kleinste gemeinsame Nenner für terminfo ist "dumb".

8.6.10 Die Konsole wiederherstellen

Manchmal führen Ausgaben auf der Konsole, `$cat` *`Binärdatei`* ist ein solcher Vertreter, dazu, dass die Aus- und Eingaben unleserlich werden. Rettung verschafft hier ein blind eingetipptes:

```
$ reset
```

8.6.11 Konvertieren einer Textdatei von DOS nach Unix

Eine DOS-Textdatei, die man an ihrem Zeilenende `^M^J` erkennt, kann mit einem einzigen Kommando in eine Unix-Textdatei (Zeilenende = `^J`) gewandelt werden:

```
# apt-get install sysutils
$ dos2unix dosfile
```

8.6.12 Reguläre Ausdrücke

Alle Vorkommen des regulären Ausdrucks *FROM_REGEX* können durch *TO_REGEX* in allen Dateien *FILES* ersetzt werden durch:

```
$ perl -i -p -e 's/FROM_REGEX/TO_REGEX/g;' FILES ...
```

`-i` zeigt an, dass die Orginaldateien bearbeitet werden, `-p` sorgt ausdrücklich für die Iteration über die Dateinamen. Wenn der reguläre Ausdruck kompliziert ist, kann man sich versichern, indem man die Originaldateien behält: Durch `-i.bak` anstelle von `-i` bleiben die Originale erhalten und bekommen die Endung `.bak`.

8.6.13 Extract differences and merging updates for the source file

Following one of the procedures will extract difference of the source file and create unified diff files *file.patch0* or *file.patch1* depending on the file location:

```
$ diff -u file.old file.new1 > file.patch0
$ diff -u old/file new1/file > file.patch1
```

The diff file (alternatively called patch file) is used to send program update. Reciving party will apply this update to another *file* by:

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

If you have all 3 version of source codes, you can merge them more effectively using `diff3`:

```
$ diff3 -m file.mine file.old file.yours > file
```

8.6.14 Eine große Datei in kleine zerlegen

```
$ split -b 650m datei   # in 650MB große Stücke trennen
$ cat x* >großedatei    # wieder zusammenfügen
```

8.6.15 Kleine Tricks mit Pipes

Hier folgen einige kleine lehrreiche Beispiele zur Verwendung von Pipes:

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                     # Find alle Dateien unterhalb /usr
                     # exklusive bestimmte Pfade
xargs -n 1 command   # Starte Befehl für alle Zeilen von stdin
xargs -n 1 echo|     # Beliebige Leerzeichen werden zu Zeilenumbrüchen
grep -e pattern|     # Gebe Zeilen aus, die pattern
                     # enthalten
cut -d: -f3 -|       # gebe das dritte Feld aus, mit : als Trenner
                     # (z.B. in der Passwortdatei passwd)
col -bx |            # Entferne Backspace und expandiere Tabs zu Leerzeichen
expand -|            # expandiere Tabs zu Leerzeichen
sort -u|             # Sortieren und Duplikate entfernen
tr '\n' ' '|         # mehrere Zeilen zu einer zusammenfügen
tr '\r' ''|          # CR entfernen
tr 'A-Z' 'a-z'|      # Großbuchstaben nach Kleinbuchstaben wandeln
```

```
sed 's/^/# /'|       # Aus der Zeile einen Kommentar machen
sed 's/\.ext//g'|    # Entferne .ext
sed  -n -e 2p|       # Zeige die zweite Zeile
head -n 2 -|         # Zeige die ersten beiden Zeilen
tail -n 2 -|         # Zeige die letzten beiden Zeilen
```

8.6.16 Text oder Mailinglist aus einer Webseite extrahieren

Der Textbrowser `lynx` leistet dieses ohne Interaktion: So kann man leicht Code-Beispiele aus einer Webseite holen, ohne mühsame Dialoge.

```
$ lynx -dump http://www.remote-site.com/help-info.html >textfile
```

`links` und `w3m` haben eine ähnliche Funktion, die Ergebnisse können sich aber leicht unterscheiden

Falls ein Archiv einer Mailing-Liste geladen wird, kann `munpack` benutzt werden, um den MIME-Inhalt zu extrahieren.

8.6.17 Formatierte Ausgabe von Webseiten

Um eine Webseite zu drucken, kann diese in das PostScriptformat gewandelt werden:

```
$ apt-get install html2ps
$ html2ps URL | lpr
```

Siehe auch '`lpr/lpd`' auf Seite 43. Alternativ kann die Textdatei (siehe vorherigen Absatz) auch mit `a2ps` oder `mpage` zum Drucken aufbereitet werden.

8.6.18 Pretty print a manual page

The following will print a manual page into a PostScript file/printer.

```
$ man -Tps some-man-page | lpr
$ man -Tps some-man-page | mpage -2 | lpr
```

8.6.19 Merge two Postscript or PDF files

You can merge two Postscript or PDF files.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite \
  -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite \
  -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

8.6.20 Ausführungsdauer messen

Um die verschiedenen Zeiten, die mit einem Prozessablauf verbunden sind, anzuzeigen, kann `time` verwendet werden.

```
# time some-command >/dev/null
real    0m0.035s       # "Echte" Zeit
user    0m0.000s       # Zeit in Benutzermodus
sys     0m0.020s       # Zeit in Kernelmodus
```

8.6.21 `nice`-Befehl

Der `nice`-Befehl dient dazu, die Priorität eines Prozesses zu setzen. Dessen Verwandter `renice` (Paket `bsdutils`) ändert die Priorität eines bereits laufenden Prozesses. Letzteres kann auch aus `top` heraus geschehen. Ein `nice`-Wert von 19 bedeutet niedrigste Priorität - der Prozess ist langsam –, absteigend wird die Priorität erhöht. Kleinere Werte als 0 kann nur der Administrator setzen, -20 ist die höchste Priorität.

```
# nice  -19 top                                              # sehr nett
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # sehr schnell
```

Manchmal kann ein sehr langsamer Prozess dem System mehr schaden als nützen, also ist Vorsicht geboten.

8.6.22 Terminplanung für Prozesse (`cron`, `at`)

Mit `cron` und `at` können zu bestimmten Terminen Befehle ausgeführt werden. Siehe `at(1)`, `crontab(5)`, `crontab(8)`.

Die Tabelle für wiederkehrende Ausführung kann mit `crontab -e` bearbeitet werden. (Ein anderer Editor als `vi` kann mit der Umgebungsvariablen `EDITOR` festgelegt werden.) Beispiele einer crontab-Tabelle:

```
# Verwende /bin/sh zur Ausführung, egal was passwd sagt
SHELL=/bin/sh
# Standardausgabe wird an ‘paul’ geschickt.
MAILTO=paul
# Min Stunde TagDesMonats Monat TagDerWoche Befehl (Komma wird ’und’)
# starte um 00:05, jeden Tag
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# starte unm 14:15 am Monatsersten -- Ausgabe per Email an Paul
15 14 1  * *   $HOME/bin/monthly
# starte um 22:00 an Werktagen (1-5), schicke Email an Joe. % für Zeilenumbru
```

```
0 22 *   * 1-5 mail -s "Es ist 10 Uhr" joe%Joe,%%Gute Nacht?%.%%
23 */2 1 2 *   echo "Startet 23 Minuten nach 0 Uhr, 2 Uhr, 4 Uhr ..., am 1. F
5  4 *   * sun echo "Startet um 04:05 jeden Sonntag"
# Startet um 03:40 an jedem ersten Montag des Monats
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Um einen Befehl zur einmaligen Ausführung zu terminieren dient `at`

```
$ echo 'command -args'| at 3:40 monday
```

8.6.23 Konsolenumschaltung mit `screen`

Mit `screen` kann man mehrere virtuelle Terminals auf einer einzigen Konsole starten, jedes mit einer eigenen interaktiven Shell. Selbst wenn man immer zwischen virtuellen Terminals umschalten kann oder mehrere xterm-Fenster offen hält, sollte man sich `screen` wegen seiner vielfältigen Möglichkeiten anschauen. Dazu gehören

- Kommando-Rückholung,
- Kopieren-Einfügen,
- protokollieren,
- Eingabe von Umlauten und
- die Möglichkeit eine komplette Sitzung an einem Terminal anzuhalten und später fortzusetzen.

Beispiel für einen Fernzugriff

Wenn man sich häufig mit einem VT100-Terminal-Programm über ein Netzwerk auf einem PC einwählt, wird man das Sitzungsmanagement mit `screen` sicherlich nützlich finden.

1 Angenommen man ist über eine Telefonverbindung eingeloggt, eine `screen`-Sitzung ist gestartet und man hat einen längeren Text mit einem Editor geschrieben. Aus irgendwelchen Gründen, muss man jetzt die Verbindung unterbrechen.
2 Durch Drücken von `^A d` wird die aktuelle Sitzung gesichert. (Noch schneller geht es mit `^A DD`, dann wird sogar gleich der Logout für einen vorgenommen.)
3 Wenn man sich später wieder einwählt kann man mit `screen -r` die Sitzung holen, und `screen` wird die Sitzung so restaurieren, wie sie verlassen wurde.

Typische `screen`-Befehle

Wenn der `screen`-Prozess gestartet ist, reicht dieser alle Tastatureingaben an das aktuelle Fenster, bis auf die Kommandosequenz, die auf `^A` voreingestellt ist. Alle `screen`-Befehle beginnen mit `^A` und werden dann von einem einzelnen Buchstaben gefolgt (ggf. folgen dann die Parameter). Beispiele für Befehle:

```
^A ?     Hilfe (Tastaturbelegung)
^A c     Neues Fenster und wechsele dorthin
^A n     Gehe zum nächsten Fenster
^A p     Gehe zum vorherigen Fenster
^A 0     Gehe zum Fenster Nummer 0
^A w     Zeige eine Liste der Fenster
^A a     Gebe das ^A an die Applikation weiter
^A h     Hardcopy des Fensters in Datei
^A H     Starte/Beende Protokoll des aktuellen Fensters in eine Datei
^A ^X    Sichere das Terminal mit einem Passwort
^A d     Aktuelle Sitzung vom Terminal abmelden
^A DD    Vom Terminal abmelden und Logout
```

Dies ist nur eine kleine Auswahl der Befehle, die das mächtige `screen` unterstützt. Siehe `screen(1)` für weitere Details.

Backspace und/oder Strg-H in einer `screen` Sitzung

Wenn die Backspace- oder Strg-H-Taste während einer `screen`-Sitzung nicht funktionieren, muss die Zeile

```
bindkey -k kb stuff "\177"
```

in der Datei `/etc/screenrc` auskommentiert ("#") werden.

`screen`-Äquivalent unter X

Siehe Paket `xmove` und dort `xmove(1)`.

8.6.24 Grundlagen - Prüfung des Netzwerks

Installiere um die grundlegenden Funktionen des Netzwerks zu überprüfen `netkit-ping`, `traceroute`, `dnsutils`, `ipchains` (für 2.2 Kernel), `iptables` (für 2.4 Kernel), und das `net-tools` Paket und führe folgende Sequenz aus:

```
$ ping yahoo.com            # teste Internetverbindung
$ traceroute yahoo.com      # tracen der IP-Pakete
$ ifconfig                  # testen der host-Konfiguration
$ route -n                  # testen der routing-Konfiguration
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # teste host.dom DNS-Einträge am dns-server.com
      # und suche dort nach {mx|any} Eintrag
$ ipchains  -L -n |less     # teste packet-Filter (2.2 kernel)
$ iptables -L -n |less      # teste packet-Filter (2.4 kernel)
```

```
$ netstat -a                # Finde alle offenen Ports
$ netstat -l --inet         # Finde alle auf Eingabe wartenden Ports
$ netstat -ln --tcp         # Ebenso (TCP, numerisch) Ports
```

8.6.25 Email aus dem lokalen Spooler leiten (flush)

Um Email aus dem lokalen Spooler weiterzuleiten:

```
# exim -q    # wartende Email anstoßen
# exim -qf   # alle Emails weiterleiten
# exim -qff  # dasselbe, auch Email mit Status 'frozen' wird weitergeleitet
```

`-qff` wäre wohl die bessere Wahl in dem Skript `/etc/ppp/ip-up.d/exim`.

8.6.26 'Frozen' Email aus dem lokalen Spooler entfernen

Um nicht weiter vermittelbare Email aus dem lokalen Spooler zu entfernen und eine Fehlermeldung zurückzugeben dient:

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

8.6.27 Re-deliver mbox contents

You need to manually deliver mails to the sorted mail boxs in your home directory from `/var/mail/`*`username`* if your home directory became full and procmail failed. After making disk space in the home directory, run:

```
# /etc/init.d/exim stop
# formail -s procmail </var/mail/username
# /etc/init.d/exim start
```

8.6.28 Dateiinhalte zurücksetzen

Um eine Datei zurückzusetzen, z.B. ein Log-Datei, sollte nicht `rm` benutzt werden, da diese Dateien im Sekundenrhythmus geschrieben werden. Sicherer ist:

```
$ :>dateiname
```

8.6.29 Dummy-Dateien

Die folgenden Befehle erzeugen leere Dateien beliebiger Größe:

```
$ dd if=/dev/zero    of=dateiname bs=1k count=5 # 5KB große, aber leere Datei
$ dd if=/dev/urandom of=dateiname bs=1m count=7 # 7MB mit Zufallsinhalt
$ touch dateiname #erzeuge 0B Datei (wenn sie existiert, setze mtime neu)
```

8.6.30 `chroot`

Wenn man neben seiner Debian-Distribution noch eine andere installiert hat (dies kann auch jede andere Distribution sein), z.B. auf der Partition */dev/hda1*, dann kann dieses zweite System mit dem aktuellen Kernel ohne Warmstart gestartet werden.

```
# mount /dev/hda1 /mnt/target
 ... dies nimmt an, dass das zweite System auf /dev/hda1 liegt
# chroot /mnt/target
 ... Jetzt wird der Inhalt von /dev/hda1 als root-Verzeichnis betrachtet
# mount proc /proc              # reine Vorsicht
 ... Ausgeführte Befehle stammen nun von  /dev/hda1
```

So kann man verschiedene Stabilitätszweige (stable/testing/unstable) auf einer Maschine testen. Eine weitere Möglichkeit wäre, ein fremdes System per NFS zu mounten und `chroot` auf dieses neu gemountete Verzeichnis anzuwenden.

Ein auf `chroot` basierendes System kann leicht mit `debootstrap` (in Woody) erzeugt werden.

```
# mkdir potatochroot
# debootstrap potato potatochroot
# chroot potatochroot
# apt-setup # set-up /etc/apt/sources.list
```

Es gibt ein ausgefeilteres `chroot`-Paket, `pbuilder`. Es erzeugt eine `chroot`-Umgebung und stellt ein Debian-Paket in dieser Sandbox zusammen. So kann man Paketabhängigkeiten definieren, erkennen und aufzulösen.

See http://people.debian.org/~blade/install/

8.6.31 How to check hard links

You can check whether 2 files are the same file with 2 hard links by:

```
$ ls -li file1 file2
```

8.6.32 Ein Festplatten-Image `mounten`

Wenn *`file.img`* ein Festplatten-Image enthält und die Quellfestplatte eine Konfiguration wie *xxxx* = (Byte/Sektor) * (Sektor/Zylinder) hat, dann kann dieses Image mit dem folgenden Befehl nach `/mnt` gemountet werden:

```
# mount -o loop,offset=xxxx file.img /mnt
```

Die meisten Festplatten haben 512 bytes/Sektor

8.6.33 Samba

Grundlagen um auf Windows-Dateien zuzugreifen:

```
# mount -t smbfs -o username=myname,uid=my_uid,gid=my_gid \
        //server/share /mnt/smb  # freigegebene Windows-Laufwerke mounten
# smbmount //server/share /mnt/smb \
        -o "username=myname,uid=my_uid,gid=my_gid"
# smbclient -L 192.168.1.2 # Freigaben anzeigen
```

Die Netzwerk-Nachbarschaft kann angezeigt werden mit:

```
# smbclient -N -L eigene_IP_Address | less
# nmblookup -T "*"
```

Kapitel 9

Tunen eines Debian-Systems

Dieses Kapitel beschreibt nur die Grundlagen der Systemkonfiguration mittels textbasierter Schnittstellen. Sinnvolles Vorwissen liefert 'Debian System-Installationshinweise' auf Seite 29.

Für diejenigen Leser, die besonders Wert auf die Sicherheit ihres Systems legen, sollte das Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`) Pflichtlektüre sein. Dieses Dokument findet sich auch im Paket `harden-doc`.

9.1 Hinweise zur System-Inititalisierung

Der Abschnitt 'Das `init` Programm' auf Seite 22 beschreibt die Grundlagen der Debian init-Skriptdateien.

9.1.1 Anpassen der Init-Skripte

Die Debian Distribution verwendet das sys-V System für die Init-Skripte. Da alle Dateien unterhalb von `/etc/init.d/*` als Konfigurationsdateien angesehen werden, können sie vom Systemadministrator beliebig verändert werden. Dennoch ist der empfohlene Ansatz, die Dateien in `/etc/default/*` anzupassen.

Die Datei `/etc/init.d/rcS` dient z.B. zum Einstellen von Vorgaben, die beim Systemstart aktiviert werden sollen. Hier können Programme wie `motd` oder `sulogin` konfiguriert bzw. gestartet werden.

9.1.2 Anpassen der System-Protokollierung

Die Art und Weise der System-Protokollierung kann in der Konfigurationsdatei des `syslogd` in `/etc/syslog.conf` bestimmt werden. Das `colorize` Paket kann die Ausgaben dieses Daemons nach belieben färben, um die Übersicht zu erhöhen. Vergleichen Sie `syslogd(8)` und `syslog.conf(5)`.

9.1.3 Hardware-Optimierung

Einige Optimierungen der Hardware überlässt Debian dem Systemadministrator.

- `hdparm`
  - Optimierung des Festplattenzugriffs. Sehr effektiv.
  - Aber auch gefährlich. Die Manual-Page `hdparm(8)` sollte auf jeden Fall zuvor gelesen werden.
  - `hdparm -tT /dev/hda` testet die Geschwindigkeit der Festplatte.
  - `hdparm -q -c3 -d1 -u1 -m16 /dev/hda` aktiviert einige der Optimierungen einer IDE-Festplatte (Dies kann gefährlich sein, nicht einfach übernehmen!)

- `setcd`
  - Optimierung des Zugriffs auf CD-Geräte.
  - `setcd -x 2` verringert die Geschwindigkeit auf 2x.
  - Vergleichen Sie `setcd(1)`.

- `setserial`
  - Dieses Paket enthält eine Sammlung von Programmen zur Einstellung der seriellen Schnittstelle.

- `scsitools`
  - Dieses Paket enthält eine Sammlung von Programmen zur Einstellung der SCSI-Schnittstelle.

- `memtest86`
  - Dieses Paket enthält eine Sammlung von Programmen zum Testen des Speichers.

- `hwtools`
  - Dieses Paket enthält eine Sammlung von Programmen, die den Low-Level-Zugriff auf Einstellungen der Hardware ermöglichen:
    * `irqtune`: Ändert die IRQ-Priorität von Geräten, um höhere Prioritäten und schnelleren Zugriff zu ermöglichen (z.B. für serielle Ports, Modems). Eine Beschleunigung um den Faktor 3 ist bei seriellen Ports und Modems möglich.
    * `scanport`: Durchsucht den I/O-Adressraum von 0x100 bis 0x3ff auf der Suche nach Geräten am ISA-Bus.
    * `inb`: Ein kleines Programm das den IO-Port ausliest und den Inhalt in hexadezimaler und binärer Darstellung ausgibt.

- `schedutils`
  - Linux Scheduler-Werkzeuge. (Der Scheduler verwaltet die Zeitscheibe, mit der Prozesse Rechenzeit zugewiesen bekommen.)
  - `taskset`, `irqset`, `lsrt` und `rt` sind in diesem Paket enthalten.
  - Zusammen mit den schon beschriebenen `nice` und `renice` Programmen (die aber nicht enthalten sind), können die Scheduler-Parameter bis ins Detail verwaltet werden.

Das Mounten eines Dateisystems mit der noatime Option ist auch sehr effektiv zur Beschleunigung von Lesezugriffen auf Dateien. Vergleichen Sie `fstab(5)` und `mount(8)`.

Bestimmte Hardware kann direkt durch den Linux Kernel und das proc Dateisystem getunt werden. Vergleichen Sie 'Tunen des Kernels mittels des proc Dateisystems' auf Seite 93.

Es gibt viele hardwarespezifische Konfigurationshilfsmittel in Debian. Viele von diesen beziehen sich auf Laptops. Einige interessante in Debian verfügbare Pakete sind:

- `tpconfig` - ein Programm zur Konfiguration von Touchpads
- `apmd` - Programme für Advanced Power Management (APM)
- `acpi` - zeigt Informationen zu ACPI Geräten an
- `acpid` - Hilfsmittel zur Verwendung von ACPI
- `lphdisk` - bereitet eine Hibernation-Partition für Phoenix NoteBIOS vor
- `sleepd` - läßt ein Notebook während Inaktivität schlafen
- `noflushd` - erlaubt es inaktiven Festplatten herunterzufahren
- `big-cursor` - größerer Mauscursor für X
- `acme` - Aktiviert die "Multimedia Tasten" von Laptops
- `tpctl` - Konfigurationstools für IBM ThinkPad Hardware
- `mwavem` - Unterstützung von Mwave/ACP Modems
- `toshset` - erlaubt Zugriff auf die meisten Hardwareschnittstellen eines Toshiba Laptops
- `toshutils` - Hilfsmittel für Toshiba Laptops
- `sjog` - Programm zum Zugriff auf das "Jog Dial" (Drehrad) in Sony Vaio Laptops
- `spicctrl` - Programm für das Sony Vaio zum Einstellen der Helligkeit des LCD

ACPI ist ein moderneres Protokoll für das Power-Management als APM.

Einige dieser Pakete erfordern spezielle Kernel-Module. Diese sind in der Regel bereits in den neuesten Kernelquellen enthalten. Bei Problemen kann es notwendig sein, selbst den neuesten Patch in den Kernel einzuspielen.

9.2 Zugriffskontrolle

9.2.1 Zugriffskontrolle mit PAM und login

PAM (Pluggable Authentication Modules) erlaubt die Kontrolle von Logins.

```
/etc/pam.d/*            # PAM Konfiguration
/etc/pam.d/login        # PAM Konfiguration des Logins
/etc/security/*         # PAM Modulparameter
/etc/securetty          # Konfiguration des root-Login auf der Konsole
/etc/login.defs         # generelle Logins
```

Falls wirklich (auf eigene Gefahr) ein Login an der Konsole ohne Passwort möglich sein soll, kann die Datei `/etc/pam.d/login` wie folgt geändert werden:

```
#auth       required   pam_unix.so nullok
auth       required   pam_permit.so
```

Ähnlich funktioniert das auch mit `xdm`, `gdm`, ..., um sich ohne Passwort an der X-Konsole anzumelden.

Andererseits kann `cracklib2` installiert werden und `/etc/pam.d/passwd` wie folgt angepasst werden, um einen guten Passwortschutz zu erhalten.

```
password required       pam_cracklib.so retry=3 minlen=6 difok=3
```

Ein einmaliges Nutzen eines Login Passworts zur Zugangsaktivierung kann auch hilfreich sein. Dafür muss das `passwd` Kommando mit der Option `-e` verwendet werden. Vergleichen Sie `passwd(1)`.

Die maximale Anzahl an Prozessen kann mit `ulimit -u 1000` in der Bash-Shell gesetzt werden oder auch in der Datei `/etc/security/limits.conf` mit PAM. Andere Parameter, wie z.B. `core` können so ebenfalls gesetzt werden. Die Anfangseinstellung von `PATH` kann in `/etc/login.defs` festgelegt werden, bevor die Shell Skripte startet.

Die Dokumentation von PAM befindet sich im Paket `libpam-doc`. Das Buch *Linux-PAM System Administrator's Guide* behandelt PAM, die verfügbaren Module, usw. Die Dokumentation enthält auch *The Linux-PAM Application Developers' Guide* und *The Linux-PAM Module Writers' Guide*.

9.2.2 "Warum GNU `su` nicht die `wheel` Gruppe unterstützt"

Dieser Satz entstammt dem Ende der alten `info su` Seite von Richard M. Stallman. Man sollte sich aber nicht verwirren lassen. Die aktuelle `su` Version in Debian benutzt PAM. Somit kann der Gebrauch von `su` leicht auf eine beliebige Gruppe eingeschränkt werden. Dazu dient das Modul `pam_wheel.so`, welches in `/etc/pam.d/su` konfiguriert wird. Im folgenden Beispiel wird die `adm` Gruppe äquivalent zu der BSD Gruppe `wheel` eingerichtet und erlaubt den Gruppenmitgliedern den Gebrauch des `su` Kommandos ohne Passwort.

```
# anti-RMS Konfiguration in /etc/pam.d/su
auth       required   pam_wheel.so group=adm

# wheel Mitglieder die su ohne Passwort verwenden können
auth       sufficient pam_wheel.so trust group=adm
```

9.2.3 Bedeutung verschiedener Gruppen

Interessante Gruppen sind:

- Die `root` Gruppe entspricht der Standard wheel Gruppe für `su`, wenn `pam_wheel.so` ohne das `group=` Argument verwendet wird.
- Die `adm` Gruppe darf Log-Dateien lesen.
- Die `cdrom` Gruppe darf auf das CD-ROM-Laufwerk zugreifen.
- Die `floppy` Gruppe darf auf das Diskettenlaufwerk zugreifen.

- Die `audio` Gruppe darf auf ein vorhandenes Audiogerät zugreifen.
- Die `src` Gruppe darf auf Quelltexte unterhalb von `/usr/src` zugreifen. Mit dieser Gruppe verwaltet man den Zugriff auf Quelltexte.
- Die `staff` Gruppe ist gedacht für Hilfsadministratoren. Sie sollen Dateien in `/usr/local` ablegen und Verzeichnisse in `/home` erzeugen dürfen.

Eine vollständige Liste bietet der "FAQ"-Abschnitt in Securing Debian Manual (`http://www.debian.org/doc/manuals/securing-debian-howto/`), dieses Dokument befindet sich im `harden-doc` Paket in Woody. Auch das neue `base-passwd` (>3.4.6) enthält eine maßgebliche Liste: `/usr/share/doc/base-passwd/users-and-groups.html`.

9.2.4 `sudo` – eine sicherere Arbeitsumgebung

Ich verwende `sudo` hauptsächlich als Selbstschutz. `sudo` ist die etwas sicherere Alternative zur Verwendung des Administrator-Accounts.

`sudo` wird wie üblich installiert und mittels der Datei `/etc/sudoers(examples/)` konfiguriert. Überprüfen Sie auch die Merkmale der `sudo` Gruppe in `/usr/share/doc/sudo/OPTIONS`.

Diese Beispielkonfiguration verwendet eine Gruppe "staff", welche es den Mitgliedern erlaubt, beliebige Befehle mit root-Rechten mittels `sudo` auszuführen. Desweiteren wird dieses Recht einer Gruppe "src" mit Einschränkungen auf bestimmte Befehle gegeben.

Der Vorteil von `sudo` ist, dass nur ein normales Benutzerpasswort zum Einloggen benötigt wird. Die Aktivitäten eines `sudo` Benutzers werden aufgezeichnet. Dies ermöglicht es, einige Aufgaben an Hilfsadministratoren zu delegieren. Zum Beispiel:

```
$ sudo chown -R ich:meineGruppe .
```

Selbstverständlich kann jeder der das Administratorpasswortes kennt (wie es bei den meisten Heimanwendern der Fall ist) auch auf andere Weise root-Kommandos absetzen:

```
$ su -c "shutdown -h now"
Password:
```

(Mit `sudo` können feiner abgestufte Sicherheitsbarrieren eingebaut werden. Ich hätte dies tun können. Da aber nur ich Zugriff auf meinen Rechner habe, hielt ich das nicht für nötig.)

Ein anderes Programm mit ähnlicher Funktionalität ist im `super` Paket zu finden.

9.2.5 Zugriffskontrolle auf Daemonen

Der Internet-*Superserver* `inetd` wird beim Systemboot aus `/etc/rc2.d/S20inetd` (für das Runlevel 2) gestartet; dies wiederum ist ein symbolischer Link auf `/etc/init.d/inetd`.

Durch die Verwaltung verschiedener Dienste/Daemonen in diesem einen Daemon wird die Systemlast verringert.

Wenn eine Anfrage via Netzwerk an den Rechner gestellt wird, sieht der Daemon in der Datei `/etc/protocols` und `/etc/services` nach, welches Protokoll und welcher Service nachgefragt werden. Bei einem normalen Internetservice schaut `inetd` dann in `/etc/inetd.conf` nach oder bei Sun-RPC-Anfragen in `/etc/rpc.conf`.

Die Sicherheit des Systems kann durch die Abschaltung nicht gebrauchter Services in `/etc/inetd.conf` immens gesteigert werden. Sun-RPC-Services werden gebraucht, wenn NFS oder andere RPB-basierte Protokolle verwendet werden.

Für einige Daemonen wird der gewünschte Server nicht direkt gestartet. Stattdessen wird der `tcpd` TCP/IP-Daemon-Wrapper mit dem gewünschten Server als Argument aus `/etc/inetd.conf` aufgerufen. In diesem Fall wird der Server von `tcpd` erst nach Log-Einträgen und Sicherheitsüberprüfungen gestartet (`/etc/hosts.deny` und `/etc/hosts.allow` werden ausgewertet).

Falls beim entfernten Zugriff in einem Debian-System Probleme auftreten, kann eine Ursache der Eintrag: "ALL: PARANOID" in `/etc/hosts.deny` sein, der dann auskommentiert werden muss.

Details dazu liefern: `inetd(8)`, `inetd.conf(5)`, `protocols(5)`, `services(5)`, `tcpd(8)`, `hosts_access(5)` und `hosts_options(5)`.

Weitere Informationen zu Sun-RPC liefern: `rpcinfo(8)`, `portmap(8)` und `/usr/share/doc/portmap/portmapper.txt.gz`.

9.2.6 LDAP – Leichtgewichtiges Verzeichniszugangsprotokoll

Literatur zu LDAP (Lightweight Directory Access Protocol):

- OpenLDAP (`http://www.openldap.org/`)
- OpenLDAP Admin Guide im `openldap-guide` Paket
- LDP: LDAP Linux HOWTO (`http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html`)
- LDP: LDAP Implementation HOWTO (`http://www.tldp.org/HOWTO/LDAP-Implementation-HOWTO/index.html`)
- OpenLDAP, ausführliche Nutzungsberichte (`http://portal.aphroland.org/~aphro/ldap-docs/ldap.html`)
- Open LDAP mit Courier IMAP und Postfix (`http://annapolislinux.org/docs/plc/postfix-courier-howto.txt`)

9.3 CD-Brenner

CD-Brenner mit ATAPI/IDE-Schnittstelle sind in der letzten Zeit populär geworden, da sie gut geeignet für Archivierungs- und Backupzwecke kleiner Datenmengen (< 640 MB) sind. Das

wichtigste Dokument zu diesem Thema ist das CD-Writing-HOWTO (http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html).

9.3.1 Einführung

Der Abbruch des Datenstroms zum CD-Brenner während des Brennvorgangs wird unweigerlich dazu führen, dass der Rohling unbrauchbar wird. Deshalb sollte der Puffer-Speicher so groß wie möglich sein. Wenn Geld keine Rolle spielt, sollte man lieber einen SCSI-Brenner kaufen, hat man eine Wahl bzgl. des Bussystems, sollte man den Brenner unbedingt über den PCI-Bus (z.B. auf dem Motherboard), statt den ISA-Bus (wie bei der SB16) anschließen.

Ist ein CD-Brenner am IDE-Bus angeschlossen, wird der IDE-SCSI Treiber anstatt eines gewöhnlichen IDE CD-Treibers für Linux 2.2 und 2.4 Kernel verwendet. Auch der allgemeine SCSI-Treiber muss aktiviert sein. Es gibt zwei Möglichkeiten, wenn man einen Kernel, wie er mit modernen Distributionen (März 2001) ausgeliefert wird, hat:

9.3.2 Ansatz 1: Module + `lilo`

Die folgende Zeile muss in der Datei `/etc/lilo.conf` hinzugefügt werden, wenn ein Standard-Debian-Kernel verwendet wird. Falls dort schon Einträge vorhanden sind, können diese mit Leerzeichen getrennt werden.

```
append="hdx=ide-scsi ignore=hdx"
```

Der Anschlussname eines CD-Brenners, der durch den IDE-SCSI-Treiber angesprochen wird, ergibt sich dabei wie folgt: In `hdx` ist *x* zu ersetzen gemäß:

```
hda          für den Master am ersten IDE-Anschluss
hdb          für den Slave am ersten IDE-Anschluss
hdc          für den Master am zweiten IDE-Anschluss
hdd          für den Slave am zweiten IDE-Anschluss
hde ... hdh  für ein Laufwerk an einem externen oder ATA66/100
             IDE-Anschluss.
```

Die neue Lilo-Konfiguration wird mit root-Rechten aktiviert durch:

```
# lilo
# shutdown -h now
```

9.3.3 Ansatz 2: Neukompilieren des Kernels

Debian benutzt `make-kpkg` um einen Kernel zu erstellen. Die Option `--append_to_version` ist beim Erzeugen mehrerer Kernel sinnvoll. Siehe dazu 'Der Linux Kernel unter Debian' auf Seite 87.

Nach Eingabe von `make menuconfig` geht man wie folgt vor:

- bzImage
- der IDE-CD-Treiber sollte entfernt werden
- ide-scsi und sg sollten angewählt werden (mindestens als Modul)

9.3.4 Weitere Schritte nach der Konfiguration

Die Kernel-Unterstützung für den CD-Brenner kann während des Boot-Prozesses aktiviert werden durch:

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

Manuelle Aktivierung geschieht durch:

```
# modprobe ide-scsi
# modprobe sg
```

Nach dem Neustart kann die Installation geprüft werden durch

```
$ dmesg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Von Warren Dodge] Manchmal gibt es Konflikte zwischen den `ide-scsi`- und `ide-cd`-Treibern, wenn für beide Treiber Geräte installiert sind. Durch die folgende Zeile, die zu der Datei `/etc/modutils/aliases` hinzugefügt werden soll, kann das Problem beseitigt werden. Vor dem Neustart ist noch ein Lauf von `update-modules` nötig.

```
pre-install      ide-scsi      modprobe ide-cd
```

Durch diese Zeile wird der IDE-Treiber vor `ide-scsi` geladen. Der IDE-Treiber `ide-cd` übernimmt die Kontrolle über die ATAPI-CD-ROM Geräte – über alle die nicht ignoriert werden sollen. Die anderen bisher ignorierten Geräte übernimmt dann das `ide-scsi`-Modul.

9.3.5 Bootfähiges CD-Image

Um eine bootfähige CD-Image `cd-image.raw` mit den Dateien unterhalb vom `Zielverzeichnis` zu erzeugen, geht man wie folgt vor: Man legt eine bootfähige Diskette in das erste Diskettenlaufwerk ein und erzeugt davon ein Image:

```
# dd if=/dev/fd0 Zielverzeichnis/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
          -o cd-image.raw Zielverzeichnis/
```

Die `-t` Option erzeugt die Übersetzungstabelle für lange Dateinamen, `-b` und `-c` steht für bootfähig.

Tatsächlich kann sogar eine bootfähige CD mit DOS erstellt werden. Wenn die Datei *`boot.img`* im obigen Beispiel eine DOS-Boot-Diskette beinhaltet, dann wird die CD gestartet, als ob diese Diskette im Diskettenlaufwerk stecken würde. Vielleicht möchte ja mal jemand das mit Freedos probieren.

Das CD-Image kann man mounten mit:

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6 Brennen mit dem CD-Brenner (R, RW):

Ein erster Test mit einem Double-Speed-Laufwerk sieht so aus (dabei wird keine CD gebrannt, sondern nur simuliert):

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Falls dies erfolgreich ist, kann die CD-R gebrannt werden:

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Eine wiederbeschreibbare CD-RW muss zudem erst gelöscht werden:

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Manche CD-RW-Laufwerke funktionieren besser mit:

```
# nice --10 cdrecord -v blank=all speed=2 dev=0,0 disk.img
```

gefolgt von:

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Diese getrennten Schritte sollen SCSI-Timeouts während des Löschens vorbeugen, die den Brenn-Prozess stören oder sogar unterbrechen können. Der `nice`-Wert kann noch der Systemlast angepasst werden.

9.3.7 Eine Abbilddatei (Image) einer CD erzeugen

Manche CD-R oder auch kommerzielle CD's haben defekte Sektoren, die eine Kopie mit `dd` verhindern (die Windows 98 CD ist ein Beispiel). Das `cdrecord` Paket beinhaltet den `readcd` Befehl, der dann verwendet werden kann, um CD-Images zu erzeugen. Bei einer Daten-CD ist die Vorgehensweise wie folgt: Die CD wird gemountet, `mount` zeigt dann die Anzahl der Blöcke an (1 Block=1024 Bytes). Teilt man diese Zahl durch 2, so erhält man die Anzahl der CD-Blöcke, die 2048 Bytes groß sind. `readcd` liest also die CD z.B. mit folgenden Parametern:

```
# readcd dev=Ziel,lun,scsibusnr # wähle Funktion 11
```

Hier sind alle 3 Kommandozeilenparameter meistens 0. Gewöhnlich ist die Anzahl der Sektoren, die von `readcd` zurückgegeben wird, zu hoch! Dann sollte man die Sektorzahl, die man aus dem `mount` Befehl gewonnen hat, verwenden.

Es sollte bemerkt werden, dass die Verwendung von `dd` einige Probleme bei CD-ROMs hat. Der erste Lauf von `dd` kann eine Fehlermeldung erzeugen und ein kürzeres Image mit unvollständigem Ende zur Folge haben. Der zweite Lauf von `dd` kann zu einem größeren Image mit am Ende angehangenen falschen Daten führen. Dies ist bei einigen Systemen möglich, wenn die Größe des Images nicht angegeben wird. Nur der zweite `dd` Aufruf mit der korrekten Größe ohne vorherigen CD-Auswurf nach der Fehlermeldung scheint diese Probleme zu umgehen. Wenn die von `df` bestimmte Größe des Images z.B. `46301184` Blöcke ist, muss das folgende Kommando zweimal angewandt werden, um ein korrektes Image zu erhalten (dies ist eine rein empirische Information):

```
# dd if=/dev/cdrom of=cd.img bs=2048 count=$((46301184/2))
```

9.3.8 Debian-CD-Images

Um aktuelle Informationen zu Debian-CD's zu erhalten, sollte die Debian-CD Seite (`http://www.debian.org/CD/`) besucht werden.

Wenn man eine schnelle Internet-Verbindung hat, kann man auch eine Netz-Installation durchführen:

- hierzu die Disketten-Images (`http://www.debian.org/distrib/floppyinst`)
- und ein minimales bootbares CD-Image (`http://www.debian.org/CD/netinst/`).

Wenn man keine schnelle Internet-Verbindung hat, kann man auch einfach ein Komplettpaket von einem der bekannten CD Distributoren (`http://www.debian.org/CD/vendors/`) kaufen.

Bandbreite sollte nie verschwendet werden, deswegen sollte man die Wahl des Installationsmechanismus ruhig besonnen überlegen.

Ein bemerkenswertes Projekt ist KNOPPIX – Linux Live-Dateisystem auf CD (`http://www.knopper.net/knoppix/index-en.html`). Diese CD bootet ein vollständiges Debian-System völlig ohne Installation.

9.3.9 Sicherheitskopien auf CD-R

Um grundlegende Konfigurations- und Datendateien auf CD zu sichern, kann das "backup" Skript backup (`examples/`) verwendet werden. Siehe dazu auch 'Kopieren und Archivieren eines Verzeichnisbaumes' auf Seite 100 und 'Differential backup and data synchronization' auf Seite 102.

9.3.10 Eine Musik-CD kopieren

Noch nicht ausreichend getestet ist folgendes:

```
# apt-get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

oder

```
# apt-get install cdrdao # disk at once
# cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 my_cd # CD auslesen
# cdrdao write --device /dev/cdrom --speed 8 my_cd    # auf neue CD schreiben
```

`cdrdao` erzeugt eine echte Kopie (ohne Sprünge, etc.).

9.4 Das X-Window Paket

Die graphische Arbeitsumgebung X wird von Xfree86 (`http://www.xfree86.org/`) bereitgestellt. Derzeit sind zwei Hauptvarianten bei Debian in Benutzung: Version 3.x und 4.x.

'X-Server' auf Seite 129 dies ist das Programm, das auf dem lokalen Monitor die X-Fenster und die Oberfläche zeichnet und die Tastatureingaben und Mausbewegungen auswertet.

'X-Client Programme' auf Seite 131 dies ist ein Programm, das (lokal auf einem eventuell fernen Rechner) per X-Window-Protokoll Applikationen laufen lässt.

Diese Definition kehrt den gewöhnlichen Gebrauch der Begriffe Client und Server um. Eine Erklärung hierfür liefert: `X(7)`, das LDP XWindow-User-HOWTO (`http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html`) und die Remote X Apps mini-HOWTO (`http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html`).

Es gibt verschiedene Wege, um X-basierte Programme, die lokal angezeigt werden sollen ("X-Server seitig"), auf einem fremden Rechner (einem "X-Client") zu starten.

- `xhost`
  - Host-List Mechanismus (sehr unsicher)
  - nicht-verschlüsseltes Protokoll (kann abgehört werden).
  - Diese Variante sollte möglichst nicht gewählt werden.
  - Siehe 'X-Fernverbindungen mit `xhost`' auf Seite 136 und `xhost(1x)`.

- `xauth`
  - Der MIT magic cookie-Mechanismus (unsicher, aber besser als `xhost`).
  - nicht-verschlüsseltes Protokoll (kann auch abgehört werden).
  - nur für lokal begrenzte Benutzung (weniger rechenintensiv als `ssh -X`).
  - Siehe 'Administratorrechte unter X' auf Seite 138 und `xauth(1x)`.

- `xdm`, `wdm`, `gdm`, `kdm`, ...
  - MIT magic cookie-Mechanismus (unsicher wie `xauth`).
  - Siehe `xdm(1x)` und `Xsecurity(7)` für die Erklärung der Grundlagen der X-Zugriffskontrolle
  - Siehe `wdm(1x)`, `gdm(8)`, und `kdm.options(5)` für weitere Informationen.
  - Siehe 'System-V `init` und Runlevel' auf Seite 84 um das graphische Login mit `xdm` abzuschalten, ohne das Paket zu deinstallieren.

- **`ssh -X`**
  - Port-Weiterleitung mittels secure shell (**sicher**).
  - verschlüsselt (Ressourcenverschwendung bei lokaler Anwendung)
  - nur für ferne Verbindung oder in unsicheren Netzen verwenden.
  - Siehe 'X-Fernverbindungen mit `ssh`' auf Seite 136.

Alle Netzzugänge zum X-System (außer `SSH`) erfordern TCP/IP Verbindungen. Siehe 'TCP/IP-Verbindungen mit X' auf Seite 136.

9.4.1 X-System Pakete

Es gibt einige Metapakete, um die Installation von X zu vereinfachen.

x-window-system-core Dieses Metapaket stellt die grundlegenden Komponenten des X-Window-Systems für eine Workstation bereit. Es beinhaltet Bibliotheken, einen X-Server (genauer das Paket `xserver-xfree86`), eine Reihe von Schriften und einige X-Client-Programme und Werkzeuge.

x-window-system Dieses Metapaket beinhaltet alle X-Window Komponenten, so wie sie vom XFree86-Projekt bereitgestellt wurden. Hinzu kommen einige Hilfsprogramme. Es hängt von `x-window-system-core`, `twm` und `xdm` ab. Das bedeutet, diese Abhängigkeiten werden automatisch aufgelöst und diese Pakete müssen nicht manuell zur Installation ausgewählt werden.

xserver-common-v3 In diesem Paket befinden sich die Dateien und Programme für das alte XFree86 Version 3.X (X3)

xserver-* Zusätzliche X3-Server Pakete zur Unterstützung von Hardware, die aus irgendwelchen Gründen nicht von XFree86 4.X unterstützt werden. Einige alte ATI mach64-Grafikkarten fallen darunter; einige andere Grafikkarten hängen sich mit der XFree Version in der Debian-Version woody auf, etc. (Alle Serverpakete findet man durch `apt-cache search xserver-|less`. Alle diese X3-Server hängen vom Paket `xserver-common-v3` ab.)

In den meisten Fällen allerdings sollte das Paket `x-window-system` installiert werden. (Möchte man keinen grafischen Login, sollte das Paket `xdm` entfernt werden. Eine genaue Beschreibung dieses Vorganges findet sich im Kapitel "'Let me disable X on boot!"' auf Seite 97.)

9.4.2 Hardware-Erkennung für X

Um eine automatische Hardware-Erkennung durchzuführen, braucht man folgende Pakete, die vor der X-Installation vorhanden sein sollten:

- `discover` – Hardwareerkennungssystem.
- `mdetect` – Maus-Erkennung.
- `read-edid` – Informations-Werkzeug für VESA PnP Monitore.

9.4.3 X-Server

Siehe `XFree86(1x)` für weitere Informationen zum X-Server.

Der X-Server kann direkt von der Konsole gestartet werden:

```
$ startx -- :<display> vtXX
z.B.:
$ startx -- :1 vt8 -bpp 16
... startet auf vt8 am Rechner  localhost:1 mit 16 bpp-Farbmodus.
```

Argumente nach dem `--` werden an den X-Server weitergeleitet.

Wenn eine `~/.xserverrc` Konfigurationsdatei verwendet wird, um das Starten des X-Servers genauer im Griff zu haben, muss darauf geachtet werden, dass der Server auch explizit mit `exec` gestartet wird, sonst werden Start und Beenden des Servers sehr langwierig:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Konfiguration des X4-Servers

Um den X4-Server zu konfigurieren kann `dpkg` benutzt werden.

```
# dpkg-reconfigure --priority=low xserver-common
# dpkg-reconfigure --priority=low xserver-xfree86
```

Es wird eine `/etc/X11/XF86Config-4` Datei erzeugt, die mit dem Skript `dexconf` modifiziert werden kann.

Konfiguration des X3-Servers

Um den X3-Server zu konfigurieren kann auch `dpkg` benutzt werden.

```
# dpkg-reconfigure --priority=low xserver-common-v3
# dpkg-reconfigure --priority=low xserver-mach64
```

Die Konfigurationsdatei von X3 heißt `/etc/X11/XF86Config` und wird mit `xf86config-v3` bearbeitet.

Manuelle Konfiguration von X4

Um Benutzer spezifische Einstellungen vorzunehmen, darf nicht der Text verändert werden, der von debconf erstellt wurde, also ist der Bereich zwischen dem BEGIN und dem END tabu.

```
### BEGIN DEBCONF SECTION
[snip]
### END DEBCONF SECTION
```

Statt dessen können die Anpassungen vor dem BEGIN eingetragen werden, wie in dem folgenden Beispiel:

```
Section "Device"
  Identifier        "Custom Device"
  Driver            "ati"
  Option            "NoAccel"
EndSection

Section "Screen"
  Identifier  "Custom Screen"
  Device      "Custom Device"
  Monitor     "Generic Monitor"
  DefaultDepth 24
  Subsection "Display"
    Depth       8
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       16
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       24
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
EndSection

Section "ServerLayout"
    Identifier        "Custom"
    Screen            "Custom Screen"
    InputDevice       "Generic Keyboard" "CoreKeyboard"
    InputDevice       "Configured Mouse" "CorePointer"
EndSection
```

9.4.4 X-Client Programme

Die meisten X-Programme können einfach von der Kommandozeile aus gestartet werden:

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Die hier angeführten Parameter bedeuten:

- `-geometry` *`BREITE`*`x`*`HÖHE`*`+`*`X-OFFSET`*`+`*`Y-OFFSET`*: anfängliche Größe und Lage des neuen Fensters.
- `-fn` *`FONTNAME`*: die Schrift, die im Fenster verwendet wird. *`FONTNAME`* kann sein:

  - a14: Normale Größe
  - a24: Große Größe
  - ... (Weitere Fonts können mit `xlsfont` ermittelt werden.)

- `-display` *`displayname`*: der Name des X-Servers, der verwendet wird. *`displayname`* kann sein:
  - *`hostname`*`:`*`D.S`* bedeutet Anzeige *`S`* auf dem Bildschirm *`D`* auf dem Rechner *`hostname`*; der X-Server für dieses Display liegt auf dem TCP-Port 6000+D.
  - *`host`*`/unix:`*`D.S`* bedeutet Anzeige *`S`* auf dem Bildschirm *D* auf dem Rechner *`host`*; der X-Server für diese Anzeige lauscht auf dem Unix-Domain-Socket `/tmp/.X11-unix/XD` (so ist es nur auf dem Rechner *`host`* erreichbar).
  - *`:D.S`* bedeutet das gleiche wie *`host`*`/unix:`*`D.S`*, wobei *`host`* der Name des lokalen Rechners ist.

Der Standard *`displayname`* für die X-Client-Programme kann mit der DISPLAY Umgebungsvariablen gesetzt werden. Bevor man nun ein X-Programm startet kann die Umgebungsvariable z.B. wie folgt gesetzt werden:

```
$ export DISPLAY=:0
        # Die Standardeinstellung
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

Der Start kann in der Datei `~/.xinitrc` konfiguriert werden. Zum Beispiel:

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

Wie in 'Angepasste X-Sitzung' auf der nächsten Seite beschrieben, haben diese Einstellungen Vorrang vor denen von Xsession, welches bei `startx` verwendet wird. Diese Methode sollte nicht verwendet werden.

9.4.5 X-Session

Eine X-Session kann auf folgende Weisen gestartet werden:

- `startx`: ein Skript, das `initx` und so den X-Server und Client auf der Konsole startet. Wenn die Datei `~/.xinitrc` nicht existiert, wird `/etc/X11/Xsession` via `/etc/X11/xinit/xinitrc` ausgeführt.

- `xdm`, `gdm`, `kdm` oder `wdm`: Dies sind Daemonprogramme, die den Login von einer bereits gestarteten graphischen Oberfläche ermöglichen. `/etc/X11/Xsession`: direktes Starten.

Die Konsole erhält man mittels "'Let me disable X on boot!"' auf Seite 97.

Angepasste X-Sitzung

Im Endeffekt ist der Start durch `/etc/X11/Xsession` nur eine Kombination von `/etc/X11/Xsession.d/50xfree86-common_determine-startup` und `/etc/X11/Xsession.d/99xfree86-common_start`.

Die Ausführung von `/etc/X11/Xsession` wird beeinflußt durch `/etc/X11/Xsession.options` und ist im wesentlichen bestimmt für die Ausführung des ersten Programms, das in folgenden Skripten durch 'exec' ausgeführt wird:

1 `~/.xsession` oder `~/.Xsession`, wenn vorhanden.
2 `/usr/bin/x-session-manager`, wenn vorhanden.
3 `/usr/bin/x-window-manager`, wenn vorhanden.
4 `/usr/bin/x-terminal-emulator`, wenn vorhanden.

Was nun genau gestartet wird, hängt von den Einstellungen des Debian-Alternativen-Systems ab, wie es in 'Alternative Befehle' auf Seite 83 erklärt wird. Zum Beispiel:

```
# update-alternatives --config x-session-manager
... oder
# update-alternatives --config x-window-manager
```

Um einen anderen X-Window-Manager einzustellen als die Voreinstellung und dennoch KDE oder Gnome als Sitzungsmanager beizubehalten, muss die Datei `/etc/X11/Xsession.d/50xfree86-common_determine-startup` durch die im zweiten Fehlerreport (siehe http://bugs.debian.org/168347) ersetzt werden. (Ich hoffe, dass dies bald mit aufgeführt wird.) Dann kann die Datei `/etc/X11/Xsession.options` wie folgt geändert werden, um den X-Sitzungsmanager zu umgehen:

```
# /etc/X11/Xsession.options
#
# configuration options for /etc/X11/Xsession
# See Xsession.options(5) for an explanation of the available options.
# Default enabled
allow-failsafe
allow-user-resources
allow-user-xsession
use-ssh-agent
# Default disabled (enable them by uncommenting)
do-not-use-x-session-manager
#do-not-use-x-window-manager
```

Ohne die gezeigten Veränderungen sind die Pakete `gnome-session` und `kdebase` dafür zuständig, einen X-Sitzungsmanager bereitzustellen. Das Entfernen dieser beiden Pakete ermöglicht die Einstellung eines beliebigen X-Window-Managers.

Wenn die Datei `/etc/X11/Xsession.options` eine Zeile wie `allow-user-xsession` (ohne führende Zeichen) enthält, kann der Benutzer die Datei `~/.xsession` oder `~/.Xsession` für eine beliebige benutzerseitige Erweiterung der Datei `/etc/X11/Xsession` verwendet werden.

Der letzte Befehl in `~/.xsession` sollte die Form `exec` *`some-window/session-manager`* haben, um den Window-Manager zu starten.

Ein gutes Beispiel für eine `~/.xsession` Datei findet sich hier: `/usr/share/doc/xfree86-common/examples/xsession.gz`.

Ich benutze dies, um den Window-Manager, Bildschirmzugang und die Sprache für jeden Benutzer einzeln zu setzen. Siehe 'Start eines gewählten X-Sitzungs-/Window-Managers' auf dieser Seite, 'Administratorrechte unter X' auf Seite 138 und 'Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1)' auf Seite 150.

Benutzerspezifische X-Einstellungen können in `~/.Xresources` gespeichert werden, wohingegen sich systemweite Einstellungen in `/etc/X11/Xresources/*` befinden. Siehe `xrdb(1x)`.

Benutzerspezifische Tastaturbelegungen und Maustastenbelegungen können in `~/.xmodmaprc` definiert werden. Siehe `xmodmap(1x)`.

Start eines gewählten X-Sitzungs-/Window-Managers

Eine benutzergewählte Sitzungs-/Window-Manager-Konfiguration kann also gemäß 'Angepasste X-Sitzung' auf der vorherigen Seite in `~/.xsession` durch das Ändern der letzten Zeile eingestellt werden. (Ich mag blackbox wegen dessen Geschwindigkeit und Einfachheit):

- Voreinstellung – X-Sitzungsmanager.
  - Vergleichen Sie 'Alternative Befehle' auf Seite 83.
  - `exec /usr/bin/x-session-manager`
- Voreinstellung – X-Window-Manager.
  - Vergleichen Sie 'Alternative Befehle' auf Seite 83.
  - `exec /usr/bin/x-window-manager`
- GNOME-Sitzungsmanager. (mächtiges Paket)
  - Installieren Sie das Paket: `gnome-session`
  - `exec /usr/bin/gnome-session`
- KDE-Sitzungsmanager. (ebenso mächtiges Paket)
  - Installieren Sie das Paket: `kdebase` (oder `kdebase3` für KDE3)
  - `exec /usr/bin/kde2`

- Blackbox-Window-Manager. (leichtgewichtiges Paket).
  - Installieren Sie das Paket: `blackbox`
  - `exec /usr/bin/blackbox`

- IceWM-Window-Manager. (ebenso leichtgewichtig)
  - Installiere das Paket: `icewm`
  - `exec /usr/bin/X11/icewm`

- FVWM2-Window-Manager.
  - Installieren Sie das Paket: `fvwm`
  - `exec /usr/bin/fvwm2`

- Windowmaker-Window-Manager.
  - Installieren Sie das Paket: `wmaker`
  - `exec /usr/bin/wmaker`

- Enlightenment-Window-Manager (überladen).
  - Installieren Sie das Paket: `enlightenment`
  - `exec /usr/bin/enlightenment`

Siehe X-Window-Manager (`http://www.xwinman.org`).

KDE/GNOME-Umgebung konfigurieren

Um eine komplette KDE/Gnome-Umgebung einzurichten empfiehlt sich die Installation der folgenden Metapakete:

- KDE: installiert das `kde` Paket
- GNOME: installiert das `gnome` Paket

Es sollte ein Installationswerkzeug gewählt werden, das die Installationsempfehlungen (`recommends`) des Paketes auflösen kann; dies sind u.a. `dselect` und `aptitude`. Sie sind menügesteuert und können auch durch die automatische Abhängigkeitserfassung die Installation im Vergleich zu `apt-get` beschleunigen und erleichtern.

Wer nur ein Konsolen-Login benötigt, sollte die Login-Manager `kdm`, `gdm`, `wdm` oder auch `xdm` nicht installieren (d.h. wieder markieren zur *nicht*-Installation), die bei einer Installation von X schnell automatisch mit ausgewählt werden. Erklärung hierzu in "'Let me disable X on boot!"' auf Seite 97.

Wenn Gnome anstelle von KDE als Sitzungsmanager gewünscht wird, muss dies der `x-session-manager`, wie in 'Alternative Befehle' auf Seite 83 beschrieben, geändert werden.

9.4.6 TCP/IP-Verbindungen mit X

Da X-TCP/IP-Verbindungen nicht verschlüsselt werden, besteht bei Fernverbindungen die Gefahr des Abgehörtwerdens. Aus diesem Grund ist in Debian die Standardeinstellung so gewählt, dass TCP/IP-Socket deaktiviert ist. Eine TCP/IP-Verbindung auch zur Übertragung von graphischen Elementen kann aber sicher mit `ssh` erzeugt werden. Siehe dazu 'X-Fernverbindungen mit `ssh`' auf dieser Seite.

Die Methode, die hier im folgenden beschrieben wird, ist also nur in Umgebungen zu empfehlen, die als sicher charakterisiert werden können, wie z.B. ein hinter einer gut gewarteten Firewall agierendes Intranet oder ein kleines Heimnetzwerk. Mit dem folgenden Befehl kann die momentane X-Server-Einstellung bzgl. TCP/IP herausgefunden werden:

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Entfernen von `-nolisten` stellt das Lauschen auf dem X-Server wieder ein.

9.4.7 X-Fernverbindungen mit `xhost`

`xhost` erlaubt Zugriff auf den X-Server basierend auf dem Namen des Rechners. Dies ist sehr unsicher. Das folgende Beispiel unterdrückt den Test der Rechnernamen und erlaubt so Verbindungen von jedem Rechner aus, falls TCP/IP-Socket ebenfalls aktiviert sind. (Siehe auch 'TCP/IP-Verbindungen mit X' auf dieser Seite):

```
$ xhost +
```

Das Testen der Rechnernamen kann wieder aktiviert werden mit:

```
$ xhost -
```

`xhost` kann keine Unterscheidung auf Benutzerebene vornehmen. Desweiteren können Rechnernamen natürlich gefälscht werden.

Diese Methode sollte nicht verwendet werden, wenn man sich in einer ungesicherten Netzumgebung befindet – dies betrifft auch Wählverbindungen mittels PPP. Siehe `xhost(1x)`.

9.4.8 X-Fernverbindungen mit `ssh`

Die Verwendung von `ssh` ermöglicht die sichere Übertragung von Daten von einem lokalen X-Server zu einem Applikations-Server.

- Setze auf dem entfernten Rechner in der Datei `/etc/ssh/sshd_config` die Einträge `X11Forwarding` und `AllowTcpForwarding` auf `yes`.

- Starte den X-Server auf dem lokalen Rechner.
- Öffne ein `xterm` auf dem lokalen Rechner.
- Starte `ssh` um eine Verbindung zu dem entfernten Rechner aufzubauen.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
```

- Starte eine X-basierte Applikation auf dem entfernten Rechner.

```
loginname @ remotehost $ gimp &
```

Diese Methode erlaubt es, die graphische Ausgabe eines X-Clients anzuzeigen, als liefe er über den lokalen Unix-Domain-Socket.

9.4.9 `xterm`

Alles was mit `xterm` zu tun hat, findet man hier: `http://dickey.his.com/xterm/xterm.faq.html`.

9.4.10 X-Ressourcen Datenbank

Viele ältere X-Programme, wie z.B. xterm, verwenden die X-Ressourcen, um ihr Erscheinungsbild zu konfigurieren. Die Datei `~/.Xresources` wird verwendet um derartige Daten benutzerabhängig zu speichern. Die Datei wird automatisch während des Logins an die systemweiten Einstellungen angehängt.

Es folgen einige nützliche Einstellungen, die man in `~/.Xresources` vornehmen kann.

```
! Die Schrift in den Terminals vergrößern:
XTerm*font: 9x15

! Scrollbalken anzeigen
XTerm*scrollBar: true

! Puffergröße einstellen
XTerm*saveLines: 1000
```

Diese Einstellungen können auch sofort aktiviert werden durch:

```
xrdb -merge ~/.Xresources
```

9.4.11 Administratorrechte unter X

Wenn ein graphisches Programm mit root-Rechten gestartet werden muss, müssen einige Vorkehrungen getroffen werden. Keinesfalls sollte man als Superuser einen X-Server starten, wegen der beträchtlichen Sicherheitsrisiken.

Man startet den X-Server als gewöhnlicher Benutzer und öffnet ein `xterm` Fenster. Dann folgen:

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

Wenn man diesen Trick verwendet und sich mit `su` als ein Nicht-root-Benutzer anmeldet, muss darauf geachtet werden, dass die Datei `~/.Xauthority` für diesen speziellen Benutzer Gruppen-lesbar ist.

Um diesen Vorgang zu automatisieren, kann der betreffende Benutzer eine Datei `~/.xsession` anlegen, die folgende Zeilen enthält.

```
# So funktioniert X auch wenn su benutzt wurde.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
unset XSTARTUP
# Wenn ein besonderer Window/Sitzungsmanager gewünscht wird
# kann der hier eingestellt werden.
#XSTARTUP=/usr/bin/blackbox
# This start x-window/session-manager program
if [ -z "$XSTARTUP" ]; then
  if [ -x /usr/bin/x-session-manager ]; then
    XSTARTUP=x-session-manager
  elif [ -x /usr/bin/x-window-manager ]; then
    XSTARTUP=x-window-manager
  elif [ -x /usr/bin/x-terminal-emulator ]; then
    XSTARTUP=x-terminal-emulator
  fi
fi
# Ausführung des Window/Sitzungsmanagers.
exec $XSTARTUP
```

Nun kann `su` (nicht `su -`) in einem `xterm` Fenster vom Benutzer gestartet werden. Auch graphische Programme, die von der Shell mit Superuser-Rechten gestartet wurden, können nun angezeigt werden. Dieser Trick funktioniert nur solange, wie die übliche `/etc`

`/X11/Xsession` ausgeführt wird. Wenn der Benutzer eine eigene Datei `~/.xinit` oder `~/.xsession` Datei anlegt, muss die oben erwähnte Umgebungsvariable `XAUTHORITY` an diese Skripte angepasst werden.

Alternativ, kann das Paket `sudo` verwendet werden, um diese Sequenz abzuarbeiten.

```
$ sudo xterm
... oder
$ sudo -H -s
```

Hier muss die `/root/.bashrc` folgendes enthalten:

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

Dies funktioniert auch, wenn das Home-Verzeichnis des Benutzers mit NFS gemountet wird, da root die Datei `.Xauthority` nicht lesen muss.

There are also several specialized packages for this purpose: `kdesu`, `gksu`, `gksudo`, `gnome-sudo`, and `xsu`. Some other methods can be used to achieve similar results: creating a symlink from `/root/.Xauthority` to the user's corresponding one; use of the script sux (`http://fgouget.free.fr/sux/sux-readme.shtml`); or putting "`xauth merge ~USER_RUNNING_X/.Xauthority`" in the root initialization script.

See more on the debian-devel mailing list (`http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html`).

9.4.12 TrueType-Schriften unter X

Der Standard-X-Font-Server `xfs` in Free86-4 kommt problemlos mit TrueType Schriften zurecht. Wenn man XFree86-3 verwendet, braucht man zusätzlich den Font-Server `xfs-xtt`.

Wenn man selbst Programme kompiliert, muss man darauf achten, diese gegen libXft oder libfreetype zu linken. Wenn man nur die Pakete aus der Distribution verwendet, gibt es weiter nichts zu beachten.

Zu den neuen Schriften müssen die Dateien `fonts.{scale,dir}` erzeugt werden, da diese nötige Verwaltungsinformationen für den X-Server bereitstellen.

Da **freie** Schriften oft eingeschränkt sind, kann das Installieren von Shareware- oder kommerziellen TrueType-Schriften eine Alternative sein. Um diesen Schritt zu vereinfachen, gibt es einige Pakete:

- `ttf-commercial`
- `msttcorefonts (>1.1.0)` (Dieses Paket funktioniert in Woody seit August 2002 nicht mehr, da Microsoft die von dem Paket geladenen Schriften vom Server entfernt hat.)

Man muss schon eine gute Auswahl an Schriften haben, um das besonders frei gestaltete System mit kommerziellen Schriften zu 'verunreinigen'.

9.4.13 Web-Browser (graphische)

Es gibt eine Handvoll graphischer Web-Browser in Debian

- `mozilla` Der Mozilla-Browser (neu)
- `galeon` Mozilla-basierter Browser mit Gnome Benutzeroberfläche (neu)
- `konqueror` KDE-Browser
- `dillo` GTK-Browser
- `amaya-gtk` W3C-Referenz-Browser
- `amaya-lesstif` W3C-Referenz-Browser
- `netscape-...` (viele, alt)
- `communicator-...` (viele, alt)
- ...

Die Version des `mozilla` muss zu der `galeon` Version passen, da zweiteres die Gecko-HTML-Rendering-Engine des ersteren verwendet.

Plugins für die Browser wie `mozilla` und `galeon` können installiert werden, indem man einfach die "`*.so`" in das entsprechende Plugin-Verzeichnis kopiert (dies findet man z.B. mit `locate libnullplugin`).

Erhältliche Plugins:

- Java Plugin: Download des "J2SE" von `http://java.sun.com`.
- Flash Plugin: Download des "Macromedia Flash Player 5" von `http://www.macromedia.com/software/flashplayer/`.
- `freewrl`: VRML-Browser und Netscape-Plugin
- ...

9.5 SSH

SSH (Secure SHell) ist ein sicheres Werkzeug, um Verbindungen über das Internet herzustellen. Eine freie Version von SSH namens OpenSSH ist als `openssh` Paket in Debian enthalten.

9.5.1 Grundlagen

Zunächst müssen der OpenSSH-Server und -Client installiert werden.

```
# apt-get update && apt-get install ssh
```

In `/etc/apt/sources.list` muss ein non-US-Eintrag vorhanden sein und es darf keine Datei `/etc/ssh/sshd_not_to_be_run` bestehen.

SSH beherrscht zwei Arten von Authentisierungsprotokollen:

- SSH-Protokoll Version 1:
  - In Debian-Potato wird nur die Version 1 unterstützt.

  - Diese Version unterstützt:
    * RSAAuthentication: RSA Identitäts-Schlüssel basierte Benutzererkennung
    * RhostsAuthentication: Auf .rhosts basierende Rechnererkennung (unsicher, daher nicht aktiviert.)
    * RhostsRSAAuthentication: Authentisierung mit `.rhosts` kombiniert mit RSA Host-Schlüssel (auch deaktiviert)
    * ChallengeResponseAuthentication: RSA Challenge-Response Authentisierung
    * PasswordAuthentication: Passwortbasierte Authentisierung

- SSH Protokoll Version 2:
  - Alle Versionen seit Woody verwenden bevorzugt diese Version.
  - Diese Version unterstützt:
    * PubkeyAuthentication: Public-Key basierte Authentisierung
    * HostbasedAuthentication: `.rhosts` oder `/etc/hosts.equiv` Authentisierung kombiniert mit Public-Key Rechner Authentisierung (disabled)
    * ChallengeResponseAuthentication: challenge-response Authentisierung
    * PasswordAuthentication: Passwort basierte Authentisierung

Die unterschiedlichen unterstützten Protokolle müssen beachtet werden, wenn man von oder nach Woody migriert, da man sonst vielleicht keine SSH-Verbindungen mehr zu den gewohnten Rechnern aufbauen kann.

Siehe dazu auch `/usr/share/doc/ssh/README.Debian.gz`, `ssh(1)`, `sshd(8)`, `ssh-agent(1)` und `ssh-keygen(1)` für Details.

Die wichtigsten Konfigurationsdateien für SSH sind:

- `/etc/ssh/ssh_config`: SSH-Client Voreinstellungen. Siehe `ssh(1)`. Wichtige Einträge sind:
  - `Host`: Beschränkt die folgenden Restriktionen auf diesen Rechner bzw. dieses Rechnermuster (bis zum nächsten 'Host'-Eintrag).
  - `Protocol`: Spezifiziert die Version des SSH-Protokolls. Die Voreinstellung ist "2,1".
  - `PreferredAuthentications`: Spezifiziert die Methode mit der SSH2 authentisieren soll. Die Voreinstellung ist "hostbased,publickey,keyboard-interactive,password".
  - `ForwardX11`: Diese Option ist per Voreinstellung abgeschaltet. Sie kann über die Kommandozeilenoption "`-X`" überstimmt werden, so dass auch graphische Ausgaben übertragen werden dürfen.

- `/etc/ssh/sshd_config`: Voreinstellungen des SSH-Servers. Siehe `sshd(8)`. Wichtige Einträge sind:
  - `ListenAddress`: Spezifiziert die lokalen Adressen, auf denen `sshd` lauscht. Hier sind mehrere möglich.
  - `AllowTcpForwarding`: Voreinstellung: aus
  - `X11Forwarding`: Voreinstellung: aus.

- `$HOME/.ssh/authorized_keys`: In dieser Datei liegt die Liste der voreingestellten öffentlichen Schlüssel ('public keys'), mit denen Clients sich mit dem System verbunden haben. Siehe `ssh-keygen(1)`.
- `$HOME/.ssh/identity`: Siehe `ssh-add(1)` und `ssh-agent(1)`.

Mit folgendem Kommando kann man eine `ssh`-Verbindung aufbauen:

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # SSH version 1 erzwungen
```

`ssh` ersetzt so auf sichere und praktische Weise `telnet`.

9.5.2 Port-Umleitung – für das Tunneln mit SMTP/POP3

Um eine Weiterleitung ('pipe') einzurichten, die den Port 25 eines entfernten Servers (*remote-server*) auf den Port 4025 lokal und den Port 110 des entfernten Servers auf den Port 4110 auf dem lokalen Rechner einzurichten, kann `ssh` folgendermaßen benutzt werden:

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
          benutzername@remote-server
```

Auf diese Weise können sichere SMTP/POP3-Übertragungen realisiert werden. Man setze die Option `AllowTcpForwarding` auf `yes` in der Datei `/etc/ssh/sshd_config` auf dem entfernten Rechner.

9.5.3 Mit weniger Passwörter agieren

Man kann die sich immer wiederholende Eingabe von Passwörtern umgehen, wenn man auf das RSAAuthentication (SSH1 Protokoll)-Verfahren, bzw. PubkeyAuthentication (SSH2 Protokoll)-Verfahren zurückgreift.

Auf dem entfernten System wird dazu in der Konfigurationsdatei `/etc/ssh/sshd_config` "RSAAuthentication yes" bzw. "PubkeyAuthentication yes" gesetzt.

Dann werden Schlüssel lokal erzeugt und der dazugehörige öffentliche Schlüssel auf dem entfernten System hinterlegt.

```
$ ssh-keygen          # RSAAuthentication: RSA1 Schlüssel für SSH1
$ cat .ssh/id_rsa.pub | ssh benutzer1@remote \
       "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication: RSA Schlüssel für SSH2
$ cat .ssh/id_rsa.pub | ssh benutzer1@remote \
```

```
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication: DSA Schlüssel für SSH2
$ cat .ssh/id_dsa.pub | ssh benutzer1@remote \
        "cat - >>.ssh/authorized_keys"
```

Später kann das Passwort mit "`ssh-keygen -p`" neu gesetzt werden. Bei Problemen können vermehrt Statusinformationen ausgegeben werden mit: "`ssh -v`".

In der Datei `authorized_keys` können zu einem beliebigen Rechner Optionen hinzugefügt werden, um dort z.B. Kommandos auszuführen. Siehe `sshd(8)` für weitere Details.

SSH2 verfügt über `HostbasedAuthentication`. Damit dies funktioniert, muss auf dem Server in der Datei `/etc/ssh/sshd_config` `HostbasedAuthentication` auf `yes` gesetzt werden, desgleichen in `/etc/ssh/ssh_config` oder `$HOME/.ssh/config` auf dem Client.

9.5.4 Weitere SSH-Client-Programme

Es gibt einige freie SSH-Client-Programme auf nicht-Unix-artigen Plattformen.

Windows puTTY (`http://www.chiark.greenend.org.uk/~sgtatham/putty/`) (GPL)

Windows (cygwin) SSH in cygwin (`http://www.cygwin.com/`) (GPL)

Macintosh Classic macSSH (`http://www.macssh.com/`) (GPL) [Mac OS X beinhaltet OpenSSH; es wird gestartet mit `ssh` im Terminal]

Siehe auch SourceForge.net, site documentation (`http://www.sourceforge.net/docman/?group_id=1`), "6. CVS Instructions".

9.5.5 SSH-Agent

Der öffentliche Schlüssel kann einfach in der Datei `~/.ssh/authorized_keys` hinterlegt werden, worauf er von ssh anerkannt wird.

```
$ ssh-agent
$ # die Ausgabe in der Shell einfügen
$ ssh-add .ssh/identity
$ # oder ssh-add .ssh/id_dsa oder wie auch immer der private Schlüssel heißt
$ scp remote.host.with.public.key
```

Weitere Informationen liefern `ssh-agent(1)` und `ssh-add(1)`.

9.5.6 Problemlösungen

Wenn Probleme bei der Benutzung von `ssh` auftreten, sollte man zuerst einmal die Berechtigungen der Konfigurationsdateien mittels der `ssh` Option `-v` prüfen.

Wenn man root-Rechte hat, kann mit `-P` die Benutzung der Ports 1–1023 unterbunden werden, welche bei Firewalls Schwierigkeiten bereiten.

Wenn `ssh`-Verbindungen, die ursprünglich funktionierten plötzlich nicht gestartet werden können, hat der Systemadministrator wahrscheinlich die Datei `host_key` geändert. Nachdem man sich vergewissert hat, dass dies der Fall ist und es sich nicht um den Versuch handelt, eine Verbindung zu einem gehackten Rechner zu etablieren, kann einfach die zu dem Remote-Rechner gehörige `host_key` Zeile in der Datei `$HOME/.ssh/known_hosts` gelöscht werden. Bei der nächsten Verbindungsaufnahme wird die Zeile neu generiert.

9.6 E-Mail Programme

Die E-Mail Konfiguration eines Systems zerfällt in drei Kategorien:

- Mail-Transfer-Agenten (mail transfer agent, MTA): `exim`
- Mail-Hilfsprogramme: `procmail`, `fetchmail`, `mail`, ...
- Mail-Benutzer-Agenten (mail user agent, MUA): `mutt`

9.6.1 E-Mail Transport/Transfer Agenten (Exim)

Literatur:
- `exim-doc` und `exim-doc-html` Pakete
- http://www.exim.org/

Wenn `exim` als MTA eingesetzt wird, müssen die folgenden Dateien angepasst werden:

```
/etc/exim/exim.conf     "eximconfig" bearbeitet diese Datei
/etc/inetd.conf         durch Auskommentieren von smtp läuft exim als Daemon
/etc/email-addresses    Add spoofed source address lists
Die Filter können mittels exim -brw, -bf, -bF, -bV, ...  getestet werden.
```

Ein Fangnetz für nicht existierende Email-Adressen (Exim)

In der Datei `/etc/exim/exim.conf` (Woody oder später), kann am Ende des Abschnittes DIRECTORS (gleich nach localuser: director) ein Ausdruck eingefügt werden, der auf alle Email-Adressen passt, die vorher nicht aufgelöst werden konnten (nach Miquel van Smoorenburg):

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

Wer eine komplexere Anordnung von virtuellen Domänen angelegt hat, wird an das Ende der Datei `/etc/exim/exim.conf` einen Eintrag wie folgt vornehmen: (von mir, nicht sehr ausführlich getestet):

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Dazu benötigt man einen "*" Eintrag in der Datei `/etc/email-addresses`.

Selektive Adressen-Umschreibung für ausgehende Email (Exim)

Mit `exim` kann auch die From: Zeile der ausgehenden Emails manipuliert werden. Auch dies kann in der `exim` Konfigurationsdatei `/etc/exim/exim.conf` erledigt werden:

```
*@host1.something.dyndns.org \
  "${if eq {${lookup{$1}lsearch{/etc/passwd}{1}{0}}} {1}  \
   {$0}{$1@somethig.dyndns.org}}"  frFs
```

Dies schreibt alle Adressen um, die auf den Ausdruck `*@host1.something.dyndns.org` passen.

1 Zuerst wird in `/etc/password` nachgesehen, ob der Ausdruck $1 auf einen lokalen Benutzer zutrifft.

2 Wenn dies der Fall ist, wird daraus das, was es vorher schon war ($0).

3 Wenn es den Benutzer lokal nicht gibt, wird der Domänenanteil ersetzt.

9.6.2 Mail-Hilfsprogramme (Fetchmail)

`fetchmail` wird im Daemon-Modus gestartet und ist für die Abholung von Email von POP3-Accounts von einem ISP zuständig, um die Mails dann in das lokale Email-System einzufügen. Die Konfiguration geschieht über die Dateien:

```
/etc/init.d/fetchmail
/etc/rc?.d/???fetchmail startet update-rc.d fetchmail Default Priorität 30
/etc/fetchmailrc        Konfigurationsdatei (chown 600, Besitzer ist fetchmai
```

In der alten Debian-Version Potato war das Starten von fetchmail aus `init.d` heraus noch schwierig. Dies ist mit Woody behoben. Beispiele sind außerdem zu finden in den Dateien `/etc/init.d/fetchmail` und `/etc/fetchmailrc` unter der Adresse example scripts (`examples/`).

Wenn im Email-Kopf ^M Zeichen vorkommen, sollte "stripcr" in den Optionen der Datei `$HOME/.fetchmailrc` hinzugefügt werden:

```
options fetchall no keep stripcr
```

9.6.3 Mail-Hilfsprogramme (Procmail)

`procmail` ist ein Programm zur lokalen Mail-Zustellung und kann auch als Filter verwendet werden. Die Datei `$HOME/.procmailrc` wird für jeden Benutzer benötigt. Beispiel: _procmailrc (`examples/`)

9.6.4 Email-Programme (Mail user agent) am Beispiel Mutt

Ein beliebtes Email-Programm für die Konsole ist `mutt` in Kombination mit `vim`. Einstellungen liegen in `~/.muttrc`; z.B.

```
# diese Einstellungen modifizieren
# das Verhalten des Editors.
set editor="vim -c 'set tw=72 et ft=mail'"
#
# header weeding taken from the manual (Sven's Draconian header weeding)
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Um HTML-Email oder MS-Word-Dokumente als Text anzeigen zu können, muss `antiword` installiert sein und folgende Zeilen an entweder `/etc/mailcap` (global) oder `$HOME/.mailcap` (per User) angehängt werden.

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

9.7 Lokalisation und Sprachen

Debian ist eine internationale Distribution, die eine wachsende Anzahl an Sprachen und regionaler Besonderheiten berücksichtigt. Die folgenden Abschnitte zeigen, in welcher Form Debian dies bewerkstelligt und an welchen Stellen eine Lokalisierung eingestellt werden kann. Dies betrifft in erster Linie Ein- und Ausgaben in der jeweiligen Muttersprache, Daten, numerische und finanzielle Formate, die von System zu System unterschiedlich sein können.

9.7.1 Grundlagen

Es folgen die wichtigsten Anpassungen für die Lokalisierung.

Tastatur

Debian stellt Tastaturbelegungen für nahezu zwei Dutzend Tastaturvarianten bereit. In Woody, können diese eingestellt werden mit:

- `dpkg-reconfigure --priority=low console-data # Konsole`
- `dpkg-reconfigure --priority=low xserver-xfree86 # X4`
- `dpkg-reconfigure --priority=low xserver-common-v3 # X3`

Daten

Die große Mehrheit an Debian-Software unterstützt den Umgang mit nicht-US-ASCII-Zeichen durch die LC_CTYPE Umgebungsvariable, welche den **locale**-Mechanismus der glibc konfiguriert.

- 8-bit clean: praktisch alle Programme
- andere lateinische Buchstaben (z.B. ISO-8859-1 oder ISO-8859-2): viele Programme
- multi-byte Sprachen wie Chinesisch, Japanisch oder Koreanisch: viele neue Programm

Anzeige

X-Windows kann jede Zeichen-Kodierung anzeigen, inklusive UTF-8 und unterstützt viele Schrifttechnologien. Die Unterstützung reicht über 8-bit-Schriften über 16-bit-Schriften für Chinesisch, Japanisch oder Koreanisch. Multi-byte-Zeichen werden über den XIM-Mechanismus unterstützt. Siehe ‘Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1)’ auf Seite 150.

Japanischer EUC-Kode kann auch (mit dem Paket `kon2`) auf der (S)VGA-Konsole angezeigt werden. Eine neue Möglichkeit, welche die FB-Konsole verwendet ist `jfbterm`. In diesen Umgebungen muss allerdings die Applikation selbst die Unterstützung dieser Sprachen mitbringen. `egg` ist ein Paket für Emacs und `jvim` ein japanisiertes Vim.

Übersetzungen

Viele Text-Nachrichten und Dokumente für das Debian-System liegen übersetzt vor, z.B. Fehlermeldungen, Standardausgaben, Menüs und Manual-Pages. Derzeit gibt es Projekte zur Übersetzung in: Deutsch, Spanisch, Finnisch, Französisch, Ungarisch, Italienisch, Japanisch, Koreanisch und Polnisch. Hierzu muss natürlich das entsprechende Paket in der Form `manpages-LANG` installiert sein. (*LANG* ist hier ein eindeutiger Sprachkennzeichner gemäß den zweibuchstabigen ISO-Länderkodes.) Um herauszufinden, welche Pakete zur Auswahl stehen, kann `apt-cache search manpages-|less` verwendet werden.

Um eine NLS Manual-Page anzuzeigen, muss die Umgebungsvariable LC_MESSAGES gesetzt sein. Um z.B. deutsche Manual-Pages zu sehen, muss `LC_MESSAGES=de` gesetzt sein. Das `man` Programm wird die Manual-Page dann in `/usr/share/man/de/` suchen.

9.7.2 Lokalisierung mit locale

Debian unterstützt die **locale** Technologie. Locale ist ein System zur Ausgabe von Zeichen, Daten und Zeiten und Währungseinheiten entsprechend lokaler Gegebenheiten. Es verwendet Umgebungsvariablen, um die Region festzustellen. Wenn z.B. englische und deutsche locale Pakete installiert sind, werden die Fehlermeldungen mehrsprachig.

```
$ LANG="en_US" cat foo
cat: foo: No such file or directory
$ LANG="de_DE" cat foo
cat: foo: Datei oder Verzeichnis nicht gefunden
```

Diese Funktionalität wird von der glibc bereitgestellt. (`localeconf` erlaubt eine komfortable Konfiguration der locale-Einstellungen.) Siehe auch `locale(7)`.

9.7.3 Aktivierung der Lokalisierung

Debian hat nicht alle verschiedenen Sprachen einkompiliert. Die Datei `/usr/lib/locale` zeigt, welche Sprachen sofort verfügbar sind (neben der Standardeinstellung “C”). Wenn die benötigte Sprache fehlt, gibt es zwei Möglichkeiten:

- Hinzufügen der gewünschten Locale zu `/etc/locale.gen`, anschließend `locale-gen` starten, um die Locale zu erzeugen. Siehe dazu `locale-gen(8)` und die Manual-Pages im “SEE ALSO”-Abschnitt dieser Manual-Pages.
- Starte `dpkg-reconfigure locales`, um das `locales` Paket neu zu konfigurieren. Auch bei Neuinstallation des Paketes `locales` wird das Konfigurationstool gestartet, so dass alle Einstellungen vorgenommen werden können.

9.7.4 Eine Locale aktivieren

Die folgenden Umgebungsvariablen werden ausgewertet, um die lokalen Formate zu berücksichtigen.

1 LANGUAGE: Diese Umgebungsvariable enthält eine durch Doppelpunkte separierte Liste von Sprachen (mit absteigender Priorität). Sie wird nur verwendet, wenn die POSIX-locale nicht "C" ist [in Woody; in Potato hat LANGUAGE immer Priorität vor POSIX]. (GNU-Erweiterung)

2 LC_ALL: Wenn dieser Eintrag nicht leer ist, wird er für alle Kategorien verwendet. (POSIX.1) Voreinstellung ist "" (null).

15 LC_*: Wenn dieser Eintrag nicht leer ist, wird er für die entsprechende Kategorie gesetzt. (POSIX.1). Voreinstellung "C".

Mögliche LC_*-Variablen sind:

- LC_CTYPE: Buchstaben-Klassifikation und Groß-/Kleinumschaltung.
- LC_COLLATE: Collation-Reihenfolge. (???)
- LC_TIME: Datum- und Zeit-Format.
- LC_NUMERIC: Zahlenformat.
- LC_MONETARY: Geld-Format.
- LC_MESSAGES: Format der Meldungen und Diagnostik-Anzeigen, auch für interaktive Eingaben.
- LC_PAPER: Papier-Format.
- LC_NAME: Namen-Format.
- LC_ADDRESS: Adressen-Format und Orts-Information.
- LC_TELEPHONE: Telefonnummer-Format.
- LC_MEASUREMENT: Maßeinheiten (Metrisch oder andere).
- LC_IDENTIFICATION: Metadaten über die locale-Einstellung.

16 LANG: Wenn diese nicht leer ist und LC_ALL nicht definiert ist, wird dieser Wert für alle bis dahin nicht definierten Variablen gesetzt. (POSIX.1) Voreinstellung "C".

Einige Programme ignorieren allerdings diese Variablen. In der Regel haben diese Programme ein eigenes Sprachen-Management (z.B. Netscape Navigator 4).

Das `locale`-Programm kann die aktiven locale-Einstellungen sowie andere installierte locale anzeigen. Siehe `locale(1)`. (NOTIZ: `locale -a` zeigt alle locales an, die das System kennt; das bedeutet *nicht*, dass sie verwendbar sind (weil unübersetzt). 'Aktivierung der Lokalisierung' auf der vorherigen Seite.)

9.7.5 ISO 8601 Datum-Format-locale

Die Unterstützung für die Darstellung des internationalen Datums wird durch die locale `en_DK` – **Englisch in Dänemark** – in Form des ISO 8601-Standards (`yyyy-mm-dd`) geleistet. Dies scheint aber nur für die Konsole mit `ls` zu funktionieren.

9.7.6 Beispiel für die USA (ISO-8859-1)

Zu der Datei `~/.bash_profile` folgendes hinzufügen:

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
```

9.7.7 Beispiel für Frankreich mit Euro-Symbol (ISO-8859-15)

Zu der Datei `~/.bash_profile` folgendes hinzufügen:

```
LANG=fr_FR@euro
export LANG
LC_CTYPE=fr_FR@euro
export LC_CTYPE
```

Die Tastatur kann auf das deutsche "QWERTZ" umgestellt werden wie beschrieben in 'Tastatur' auf Seite 147. Die deutschen Manual-Pages befinden sich im Paket `manpages-de`. Die rechte Alt-Taste heißt in Europa Alt-Gr. Durch Drücken dieser Taste können diverse Sonderzeichen erreicht werden. So erzeugt z.B. AltGr-E das Euro-Symbol.

Die meisten westeuropäischen Sprachen können wie beschrieben eingestellt werden.

Siehe Debian Euro HOWTO (`http://www.debian.org/doc/manuals/debian-euro-support/`) für die Konfiguration des Euro-Symbols. Das Paket XXX für weitere Details zur Nutzung deutscher Anpassungen.

9.7.8 Beispiel für ein zweisprachiges System (japanisches EUC und ISO-8859-1)

In diesem Beispiel wird ein System aufgesetzt, das japanisch und englisch unterstützt. Die Umgebungsvariable für die japanische Sprache heißt in diesem Fall ja_JP.eucJP (japanisches EUC für traditionelle japanische Unix Umgebung). Die Unterstützung betrifft X mit gleichzeitig englischen Ausgaben und Datumsangaben nach ISO-Standard. Hinzu kommt Englisch als primäre Sprache der Text-Konsole mit der Umgebungseinstellung en_US.ISO-8859-1 (ASCII mit zusätzlichen Akzenten).

- zuerst wird eine japanische Lokale ja_JP.eucJP erzeugt, wie beschrieben in 'Lokalisation und Sprachen' auf Seite 147.
- es wird ein Kana-nach-Kanji-Konverter und ein Wörterbuch installiert:
  - `canna` – Lokaler Server (NEC free-license) oder
  - `freewnn-jserver` – Netzwerk-basierter erweiterbarer Server (Public Domain)
- Installation einer Eingabemethode für Japanisch:
  - `kinput2-canna` – für X oder
  - `kinput2-canna-wnn` – für X und
  - `egg` – funktioniert mit Emacsen, optional auch im Konsole-Modus
- Japanisch-kompatibles Terminal:

  - `kterm` – X (klassisch),
  - `mlterm` – X (sehr nett, verschiedene Schriftgrößen) und

- füge japanische Schriften hinzu.

- Konfiguriere X wie beschrieben in 'Angepasste X-Sitzung' auf Seite 133. Eine an den Benutzer angepasste X-Umgebung kann in Abhängigkeit der Start-Methode von X eingestellt werden. (`startx`, `xdm`, ...)

- Letztlich wird eine `~/.xsession` Datei erzeugt, in der die Einstellungen vorgenommen werden:

```
#!/bin/sh
# This makes X work when I su to root.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi

# Japanische locale als Default, C locale als Backup
LANG=ja_JP.eucJP
export LANG
# In der Konsole wollen wir en_US.ISO-8859-1
LC_CTYPE=ja_JP.eucJP
export LC_CTYPE
# menu-Nachrichten in Englisch und ASCII :-)
LC_MESSAGES=C
export LC_MESSAGES
# aktiviere Eingabe-Methode
kinput2 &
XMODIFIERS=@im=kinput2
export XMODIFIERS
# blackbox ist ein guter Window-Manager (lightweight)
exec /usr/bin/blackbox
```

- wir fügen folgende Zeilen zu `~/.bash_profile` hinzu:

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
# Für ISO yyyy-mm-dd Datumsanzeige kommt dem japanischen mehr entgegen
LC_TIME=en_DK.ISO-8859-1
export LC_TIME
```

- folgende Zeilen können, falls mutt als Mail-Programm verwendet wird, zu `~/.muttrc` hinzugefügt werden:

```
# UTF-8 Unterstützung ist in Japan bei EMACS nicht üblich
# 7 bit Kodierung mittels iso-2022-jp ist einfacher
# Default Reihenfolge = us-ascii --> iso-8859-1 --> utf-8
#set send_charset="us-ascii:iso-8859-1:utf-8"
#set allow_8bit=yes
set send_charset="us-ascii:iso-8859-1:iso-2022-jp"
set allow_8bit=no
```

- aktiviere XIM `kinput2` für X-Applikationen
  - füge `*inputMethod: kinput2` der X resources Datei hinzu: `.~/Xresources` (teilweise automatisch in Debian)
  - Einige Programme erlauben sogar die Änderung der Kodierung während der Laufzeit (`mlterm` kann diese dynamisch mit **Strg-Mausknopf-3** ändern.
- Nun wird X gestartet: `startx` oder durch einen Displaymanager (xdm, gdm, kdm, wdm, ...)
- In X wird ein japanisch-fähiges Programm gestartet: VIM6, (x)emacs21, mc-4.5, mutt-1.4, ... (Emacs ist populär. Aber ich verwende es nicht.)
- "**Shift+Space**" schaltet die japanische Unterstützung ein und aus.

Sieh auch die SuSE Seiten für CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html).

9.7.9 Beispiel für UTF-8 in X

In Zukunft wird Unicode alle Schriftprobleme lösen. Siehe Das Unicode HOWTO (http://www.tldp.org/HOWTO/Unicode-HOWTO.html).

9.7.10 Beispiel für UTF-8 in der FB-Konsole

Die UTF-8 Unterstützung in der Konsole wird vom Paket `bterm` bereitgestellt. Der `debian-installer` verwendet dies.

9.7.11 Über locale hinaus

Wenn eine größere Anpassung an eine Sprache gewünscht wird, sollte man zunächst nachsehen, ob `tasksel` oder `aptitude` bereits eine Option für die betreffende Sprache hat. So kann eine Paketsammlung geladen werden, die bereits Dateien für die nationale Anpassung enthält, was auch für mehrsprachige Systeme Sinn macht. Wenn bei einem System mit feinjustierter Sprachunterstützung Paketkonflikte auftreten, sollte zunächst die Ursache genau festgestellt werden, damit die Spracheinstellungen nicht verloren gehen. Eventuell muss `update-alternative` verwendet werden, um den vorherigen Status wieder her zu stellen.

Die meisten größeren Programme nutzen bereits die glibc 2.2 und sind größtenteils internationalisiert. Aus diesem Grund sind lokalisierte Versionen eines speziellen Programms wie z.B. VIM überflüssig. So wird die Funktionalität von `jvim` für VIM von `vim` ab Version 6.0 in X automatisch bereitgestellt. Tatsächlich ist die Unterstützung noch etwas holprig. Da `jvim` die japanische Unterstützung eingebaut hat, sollte man sich das Programm zumindest doch einmal ansehen.

Es kann sein, dass Programme über die `locale` Konfiguration hinaus angepasst werden müssen. Das `language-env` Paket und dessen Kommando `set-language-env` erleichtern diesen Schritt.

Siehe auch die Internationalisierungsdokumente unter Einführung in i18n (`http://www.debian.org/doc/manuals/intro-i18n/`). Es richtet sich an Entwickler und Systemadministratoren.

Kapitel 10

Aufsetzen eines Gateway mit einem Debian System

Debian ermöglicht einen Universal Gateway, der mit NAT, Mail, DHCP, DNS Cache, HTTP Proxy Cache, CVS, NFS und Samba Diensten für das eigene LAN System umgehen kann. Man vergleiche Netfilter (http://www.netfilter.org/), wo viele Netzwerkkonfigurationen aufgeführt sind.

10.1 Netzwerkkonfiguration

10.1.1 Rechnerkonfiguration des Gateway

Das LAN nutzt IP Adressen eines der folgenden privaten Netzwerkbereiche, um IP Adresskollisionen mit dem Internet zu vermeiden.

```
Klasse A: 10.0.0.0                    mit Maske 255.0.0.0
Klasse B: 172.16.0.0 - 172.31.0.0     mit Maske 255.255.0.0
Klasse C: 192.168.0.0 - 192.168.255.0 mit Maske 255.255.255.0
```

Debian verwendet `/etc/network/interfaces` für die IP Konfiguration.

Wenn zum Beispiel `eth0` eine Verbindung zum Internet mit einer durch DHCP bereitgestellten IP Adresse ist und `eth1` das LAN anbindet, so sieht `/etc/network/interfaces` (bei Woody oder neuer) wie folgt aus:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

```
auto eth1
iface eth1 inet static
address 192.168.1.1
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
```

Führen Sie das folgende Kommando aus, um die neue Netzwerkkonfiguration in `/etc/network/interfaces` zu aktivieren:

```
# /etc/init.d/networking restart
```

Zur Erinnerung: Die `/etc/network/interfaces` Datei in Woody und nachfolgenden Ausgaben ist mit Potato nicht kompatibel. (Die selbe Situation kann zwischen Woody und Sarge entstehen.)

Wird eine PCMCIA NIC (Netzwerkkarte) genutzt, muss das Netzwerk in Potato durch `/etc/pcmcia/network.opts` spezifiziert werden. Im Woody System wurde dieses Problem gelöst.

Im Zweifel sollten die Ausgaben folgender Kommandos geprüft werden:

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg | more
```

Viele neue ADSL Dienstleister nutzen PPPoE für IP Verbindungen. Das Woody Installationsprogramm hat kein Menü für PPPoE, enthält aber alle benötigten Pakete, d.h. `pppoe` und `pppoeconf`. Starten Sie `pppoeconf` in der Konsole für ein einfaches Einrichten während der Installation. Manchmal gibt es bei DSL (PPPoE) MTU bezogene Probleme. Vergleichen Sie das LDP DSL-HOWTO (`http://www.tldp.org/HOWTO/DSL-HOWTO/`).

Haben Sie Probleme mit einigen Webseiten, schauen Sie in 'Eigenartige Probleme beim Zugriff auf einige Webseiten' auf Seite 47.

10.1.2 Anhaltspunkte zur Netzwerkkonfiguration

Typische Programme:

```
# apt-get install nfs samba dhcpd dhcp-client bind squid procmail
# apt-get install fetchmail ssh cvs
```

Überprüfen Sie die folgenden Dateien:

```
/etc/init.d/dhcpd      (so bearbeiten, dass nur LAN bedient wird)
/etc/host.allow        (ALL: 192.168.0.0/16 127.0.0.0/8) für NFS
/etc/exports           (nötig für NFS)
/etc/bind/db.192.168.1 (hinzufügen)
/etc/bind/db.lan       (hinzufügen)
/etc/bind/named.conf   (bearbeiten)
/etc/resolv.conf       (bearbeiten)
/etc/hosts
/etc/dhcpd.conf        (bearbeiten für LAN = eth1)
/etc/dhclient.conf     (bearbeiten zum Erzwingen von lokalen DNS)
/etc/samba/smb.conf
/etc/exim/exim.conf
/etc/mailname
/etc/aliases
/etc/squid.conf        (hinzufügen aller IPs der LAN Rechner)
```

`bind` erzeugt einen lokalen Cache DNS Server und ändert DNS zu localhost. Überprüfen Sie `/etc/resolv.conf`:

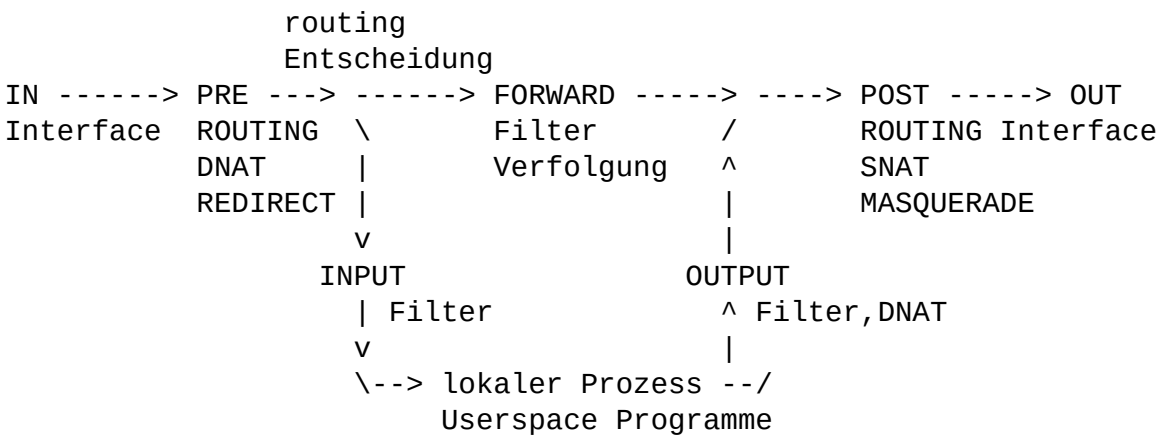
```
nameserver 127.0.0.1
search lan.aokiconsulting.com
```

10.2 Netfilter Konfiguration

Das netfilter/iptables Projekt ist ein Firewall System für Linux 2.4 und neuer. In Netfilter (http://www.netfilter.org/) werden viele Netzwerkkonfigurationen erklärt.

10.2.1 Grundlagen von netfilter

Netfilter bearbeitet Pakete unter Verwendung von fünf eingebauten Ketten: PREROUTING, INPUT, FORWARD, OUTPUT und POSTROUTING.



10.2.2 Netfilter Tabelle

Pakete werden von jeder eingebauten Kette unter Verwendung der folgenden Tabellen verarbeitet.

- filter (Paketfilter, Standard)
  - INPUT (für Pakete die im Rechner ankommen)
  - FORWARD (für Pakete die durch den Rechner geleitet werden)
  - OUTPUT (für lokal erzeugte Pakete).
- nat (network address translation, Netzwerk-Adress-Übersetzung)
  - PREROUTING (für das Ändern eingehender Pakete)
  - OUTPUT (für das Ändern lokal erzeugter Pakete vor der Weiterleitung)
  - POSTROUTING (für das Ändern ausgehender Pakete)
- mangle (network address mangling, geeignet erst nach 2.4.18)
  - alle fünf eingebauten Ketten.

10.2.3 Netfilter Ziele

Firewall Regeln haben verschiedene Ziele:

- vier Basisziele:
  - ACCEPT bedeutet, dass das Paket durchgelassen wird.
  - DROP bedeutet, dass das Paket durchfällt.
  - QUEUE bedeutet, dass das Paket den Userspace passiert (falls vom Kernel unterstützt).
  - RETURN bedeutet, dass das Durchlaufen dieser Kette beendet wird und mit der nächsten Regel der vorherigen (aufrufenden) Kette fortgefahren wird.
- erweiterte Ziele:
  - LOG aktiviert Kernel logging.
  - REJECT sendet ein Fehlerpaket zurück und lässt das Paket fallen.
  - SNAT ändert die Quelladresse des Paketes und wird nur in der POSTROUTING Kette verwendet. (nur nat Tabelle)

    ```
    --to-source ipaddr[-ipaddr][:Port-Port]
    ```
  - MASQUERADE ist das selbe wie SNAT aber für dynamisch vergebene IP (dialup) Verbindungen. (nur nat Tabelle)

    ```
    --to-ports Port[-Port]
    ```
  - DNAT ändert die Zieladresse des Pakets und wird in den PREROUTING und OUTPUT, sowie benutzerdefinierten Ketten die nur von solchen aufgerufen wurden, verwendet. (nur nat Tabelle)

    ```
    --to-destination ipaddr[-ipaddr][:Port-Port]
    ```
  - REDIRECT ändert die Ziel IP Adresse, um das Paket zum aktuellen Rechner zu senden.

    ```
    --to-ports Port[-Port]
    ```

10.2.4 Netfilter Kommandos

Die grundlegenden Kommandos von `iptables` sind:

```
iptables -N Kette          # erzeuge eine Kette

iptables -A Kette \        # füge Regel zur Kette hinzu
 -t Tabelle \              # verwende Tabelle (filter, nat, mangle)
 -p Protokoll \            # tcp, udp, icmp oder all
 -s Quelladresse[/Maske] \
 --sport Port[:Port] \     # Quellport, falls -p tcp oder udp ist
 -d Zieladresse[/Maske] \
 --dport Port[:Port] \     # Zielport, falls -p tcp oder udp ist
 -j Ziel \                 # was tun, wenn es passt
 -i in-interface-name \    # für INPUT,  FORWARD, PREROUTING
 -o out-interface-name     # für FORWARD, OUTPUT, POSTROUTING
```

10.2.5 IP-Masquerading

Maschinen im LAN können auf Internet Ressourcen mittels eines Gateway zugreifen, welcher IP-Masquerading (NAT) durchführt, indem eine einzelne extern zugängliche IP Adresse geteilt wird.

```
# apt-get install ipmasq
```

Wenden Sie die Beispielregeln an, um den `ipmasq` Schutz zu verstärken. Vergleichen Sie `/usr/share/doc/ipmasq/examples/stronger/README`. Stellen Sie beim Debian kernel-image-2.4 sicher, dass die richtigen Module geladen werden. Die nötigen Konfigurationen sind in 'Netzwerk Funktionalität' auf Seite 91 zu finden.

Für Debian kernel-image-2.2, müssen Sie `Z92timeouts.rul` in `/etc/masq/rules` wie folgt editieren, um eine längere Verbindung zu entfernten Seiten sicherzustellen (gut für große E-Mails, etc.):

```
# tcp, tcp-fin, udp
# 2 h, 10 sek, 160 sek - Standard
# 1 d, 10 min, 10 min - größeres Beispiel
$IPCHAINS -M -S 86400 600 600
```

Wird auf das Netzwerk mittels einer PCMCIA Netzwerkkarte zugegriffen, so muss `ipmasq` von `/etc/pcmcia/network.opts` gestartet werden. Lesen Sie dazu `/usr/share/doc/ipmasq/ipmasq.txt.gz`.

10.2.6 Umleiten von SMTP Verbindungen (2.4)

Angenommen, Sie haben ein Notebook, welches konfiguriert ist, andere LAN Umgebungen zu nutzen und Sie möchten Ihren Mail User Agent auf dem Notebook ohne Rekonfiguration nutzen.

Das Hinzufügen der folgenden Regeln mittels des `iptables` Kommandos zum Gateway leitet die SMTP Verbindung zum Gateway um.

```
# iptables -t nat -A PREROUTING -s 192.168.1.0/24 -j REDIRECT \
       -p tcp --dport smtp --to-port 25 # smtp=25, INPUT ist offen
```

Für gründlichere Umleitungsregeln sollte das `ipmasq` Paket installiert und `M30redirect.def(examples/)` zum `/etc/ipmasq/rules/` Verzeichnis hinzugefügt werden.

10.3 Handhaben verschiedener Netzverbindungen

[FIXME] Policy routing (von Phil Brutsche <pbrutsch@tux.creighton.edu>): Vergleichen Sie das Iproute Manual (http://lartc.org/) für Details. Traffic Control (tc) mag auch interessant sein.

Umgebung:

```
eth0: 192.168.1.2/24; Gateway 192.168.1.1
eth1: 10.0.0.2/24; Gateway 10.0.0.1
Kein Masquerading auf dieser Maschine.
```

Spezielle Werte:

1 ip rule add from 192.168.1.2 lookup 1
2 ip rule add from 10.0.0.2 lookup 2
3 ip route add to default via 10.0.0.1 metric 0
4 ip route add to default via 192.168.1.1 metric 1
5 ip route add table 1 to 192.168.1.0/24 via eth0
6 ip route add table 1 to 10.0.0.2/24 via eth1
7 ip route add table 1 to default via 192.168.1.1
8 ip route add table 2 to 192.168.1.0/24 via eth0
9 ip route add table 2 to 10.0.0.2/24 via eth1
10 ip route add table 2 to default via 10.0.0.2

[FIXME] Ich habe dies niemals durchgeführt. Wie kann eine dialup Verbindung als Backup zu einer schnellen Verbindung mit automatischer Einwahl genutzt werden? Bitte senden Sie einen Patch :)

Kapitel 11

Editoren

11.1 Populäre Editoren

Linux bietet viele Alternativen für Texteditoren in der Konsole, unter anderen:

- `vim`: Mächtiger und kleiner Editor aus der BSD Welt. VI iMproved.
- `emacs`: Ultimativer und leistungsfähiger Editor aus der GNU Welt von RMS (Richard M. Stallman).
- `xemacs`: Emacs: Die nächste Generation, ursprünglich von Lucid.
- `mcedit`: GNU Editor für Anfänger. Identisch zu `mc`'s internen Editor. Vergleiche 'Editor' auf Seite 54.
- `ae`: Kleiner Standardeditor (Potato). Nicht verwenden.
- `nano`: Kleiner GNU Standardeditor (Woody). Emuliert `pico`.
- `joe`: Für altes WordStar oder TurboPascal Feeling.
- `jed`: Schneller, reichhaltiger menübasierter Editor mit Emacs Tastenzuordnungen.
- `jove`: Sehr kleiner Editor mit Emacs Tastenzuordnungen.
- `nvi`: Neuer vi. Fehler-für-Fehler kompatibel mit dem ursprünglichen vi.

Verwenden Sie `update-alternatives --config editor`, um den Standardeditor zu wechseln. Viele Programme nutzen auch die Umgebungsvariablen `EDITOR` oder `VISUAL` um zu entscheiden, welcher Editor genutzt wird. Vergleichen Sie 'Editor' auf Seite 54.

Einige bemerkenswerte X-basierte Texteditoren:

- `gvim`: Vim mit GUI (graphische Benutzerschnittstelle) (`vim` und `vim-gtk` Paket)
- `emacs`: Der eine wahre Emacs (automatische X Erkennung).
- `xemacs`: Emacs der nächsten Generation (automatische X Erkennung).

Diese X-basierten Kommandos nutzen Standardoptionen wie `-fn` *`a24`*, was das Leben für ältere Leute wie mich einfacher macht :) Vergleiche 'X-Client Programme' auf Seite 131.

11.2 Rettungseditoren

Es gibt einige Editoren die sich in `/bin` befinden. Einer von diesen sollte installiert sein, um das Editieren von Dateien zu vereinfachen, wenn auf `/usr` nicht zugegriffen werden kann.

- `elvis-tiny`: Minimaler vi Editor (`vi` zum Starten)
- `nano-tiny`: Minimaler nicht-vi Editor (`nano-tiny` zum Starten)
- `ed`: Minimaler Editor (immer vorhanden, aber schwierig zu benutzen)

11.3 Emacs und Vim

11.3.1 Vim Hinweise

Lesen Sie das "VIM - main help file" Dokument durch Drücken von <F1> während das Programm läuft.

```
<F1>           Hilfe
<Esc>          Zurück zum normalen Modus
V              Visueller Modus
i              Eingabemodus
:              Kommandozeilen Kommandos
:set tw=72     Setzen der Textbreite auf 72
<F11>          Eingabe- (Paste) Modus
:r! date -R    Einfügen des Datums entsprechend RFC-822
qa             Aufnahme der Tastenanschläge in Register a
q              Beenden der Aufnahme
@a             Ausführen der Tastenanschläge aus Register a
:edit text.txt Editieren einer anderen Datei text.txt
:wnext         Sichern der Datei und bearbeiten der nächsten
```

`q` und `@` können für einfache Makroaufnahmen und zum Abspielen genutzt werden. Um beispielsweise ein Makro zu erzeugen, das HTML Italic Tags um das Wort unter dem Cursor einfügt, kann `qii<i>^[ea</i>^[q` (wobei `^[` die ESC Taste ist) genutzt werden. Die Eingabe von `@i` beim Start eines Wortes fügt und ein.

Vergleichen Sie auch 'Benutzung von GnuPG mit Vim' auf Seite 197.

11.3.2 Emacs Hinweise

```
<F1>               Hilfe
<F10>              Menü
C-u M-! date -R    Einfügen des Datums entsprechend RFC-822
```

11.3.3 Starten des Editors

```
Starten des Editors:                    emacs Datei  vim Datei
Starten mit vi Kompatibilität:                       vim -C
Starten ohne vi Kompatibilität:                      vim -N
Starten mit einkompilierten Standards: emacs -q      vim -N -u NONE
```

11.3.4 Zusammenfassung der Editorkommandos (Emacs, Vim)

beenden:	C-x C-c	:qa /:wq /:xa /:q!
zurücksetzen/Kommandomodus:	C-g	<Esc>
zurück (links):	C-b	h
vorwärts (rechts):	C-f	l
nächstes (herunter):	C-n	j
vorheriges (hoch):	C-p	k
StArt der Zeile (^):	C-a	0
Ende der Zeile ($):	C-e	$
mehrere Kommandos:	C-u nnn cmd	nnn cmd
mehrere Kommandos:	M-Ziffer cmd	
abspeichern:	C-x C-s	:w Datei
zum Anfang des Puffers:	M-<	1G
zum Ende des Puffers:	M->	G
eine Seite vorwärts scrollen:	C-v	^F
1/2 Seiten vorwärts scrollen:		^D
eine Zeile vorwärts scrollen:		^E
eine Seite rückwärts scrollen:	M-v	^B
1/2 Seiten rückwärts scrollen:		^U
eine Zeile rückwärts scrollen:		^Y
das andere Fenster scrollen:	M-C-v	
unter Cursor löschen:	C-d	x
löschen bis zum Zeilenende:	C-k	D
inkrementelle Suche vorwärtS:	C-s	
inkrementelle Suche Rückwärts:	C-r	
vorwärtS suchen:	C-s Enter	/
Rückwärts suchen:	C-r Enter	?
inkr. Suche nach reg. Ausd.:	M-C-s	
inkr. Suche rückw. n. reg. Ausd.:	M-C-r	
suche nach regulären Ausdruck:	M-C-s Enter	/
suche rückw. nach reg. Ausd.:	M-C-r Enter	?
Hilfe:	C-h C-h	:help
Hilfe Apropos:	C-h a	
Hilfe zu Tastenzuordnungen:	C-h b	:help [key]
Info Seiten anzeigen:	C-h i	
Hilfe Hauptmodus:	C-h m	
Hilfe Tutorial:	C-h t	:help howto
rückgängig:	C-_	u
nochmals tun (redo):	C-f	^R
markiere die Cursorposition:	C-@	m{a-zA-Z}
wechseln der Markierung und Pos.:	C-x C-x	
springe zur Mark. in akt. Datei:		'{a-z}
springe zur Mark. in jeder Datei:		'{A-Z}
Bereich kopieren:	M-w	{visual}y
Bereich entfernen:	C-w	{visual}d

```
Bereich einfügen:                C-y
einfügen des gelöschten Textes: M-y             p
in Großbuchstaben umwandeln:     C-x C-u        {visual}U
in Kleinbuchstaben umwandeln:    C-x C-l        {visual}u
einfügen spezieller Zeichen:     C-q octalnum/keystroke
                                                ^V decimal/keystroke
ersetzen:                        M-x replace-string  :%s/aaa/bbb/g
ersetzen regulärer Ausdrücke:    M-x replace-regexp  :%s/aaa/bbb/g
vor Ersetzen nachfragen:         M-%                 :%s/aaa/bbb/gc
vor Ersetzen nachfragen:         M-x query-replace
vor Ersetzen reg. Aus. nachfragen: M-x query-replace-regexp
Datei öffnen:                    C-x C-f        :r file
Datei sichern:                   C-x C-s        :w
alle Puffer speichern:           C-x s          :wa
speichern unter:                 C-x C-w file   :w file
nach Puffer fragen:              C-x b
Puffer anzeigen:                 C-x C-b        :buffers
Read-only Flag wechseln:         C-x C-q        :set ro
nach Puffer fragen und löschen:  C-x k
vertikal aufteilen:              C-x 2          :split
horizontal aufteilen:            C-x 3          :vsplit (ver. 6)
in anderes Fenster wechseln:     C-x o          ^Wp
entfernen dieses Fensters:       C-x 0          :q
entfernen anderer Fenster:       C-x 1          ^Wo
Shell im Hintergrund starten:    M-x compile
Shell im Hintergrund killen:     M-x kill-compilation
make starten:                                   :make Makefile
Fehlermeldungen überprüfen:      C-x‘           :echo errmsg
Shell starten und aufzeichnen:   M-x shell      :!script -a tmp
...clean BS, ...                                :!col -b <tmp >record
...sichern/abrufen der Aufz.:    C-x C-w record :r record
Shell starten:                   M-! sh         :sh
Kommando starten:                M-! Kmd        :!Kmd
Kommando starten und einfügen:   C-u M-! Kmd    :r!Kmd
Filter starten:                  M-| Datei      {visual}:w Datei
Filter starten und einfügen:     C-u M-| Filter {visual}:!Filter
Option anzeigen:                                :se[t] {Option}?
Option auf Standard zurücksetzen:               :se[t] {Option}&
boolsche Option zurücksetzen:                   :se[t] no{Option}
boolsche Option ändern:                         :se[t] inv{Option}
Textumbruch bei Spalte 72:                      :se tw=72
keinen Zeilenumbruch:                           :se tw=0
automatische Einrückung:                        :se ai
Tabulatoren expandieren:                        :se et
Kommentar angeben (E-Mail):                     :se comments=n:>,n:\|
```

```
GDB starten                       M-x gdb
beschreibe den GDB Mode           C-h m
eine Zeile ausführen              M-s
nächste Zeile                     M-n
eine Anweisung ausführen (stepi) M-i
aktuelle Funktion beenden         C-c C-f
weiter                            M-c
zur übergeordneten Funktion       M-u
zur untergeordneten Funktion      M-d
copy number from point, insert at the end
                                  C-x &
setze einen Haltepunkt            C-x SPC
```

11.3.5 Vim Konfiguration

Um alle Vim Eigenschaften und Syntax Highlighting zu nutzen, müssen die folgenden Zeilen in `~/.vimrc` oder `/etc/vimrc` eingefügt werden:

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Der Paste Modus unterbindet Probleme mit der automatischen Einrückung bei Markieren-und-Einfügen Operationen in konsolenbasierten Terminals. Dies bewirkt mehr als ein einfaches ":set noai".

Vergleichen Sie 'Benutzung von GnuPG mit Vim' auf Seite 197 für die GnuPG Integration.

11.3.6 Ctags

Starten Sie `apt-get install exuberant-ctags` und `ctags` auf den Quelldateien. Geben Sie `:tag` *`Funktionsname`* in Vim ein, um zur Zeile zu springen, wo *Funktionsname* beginnt. Die Tags funktionieren mit C, C++, Java, Python und vielen anderen Sprachen.

Emacs besitzt die gleichen ctags Fähigkeiten.

11.3.7 Konvertierung eines mit Syntax-Highlighting versehenen Schirms in HTML

`so \$VIMRUNTIME/syntax/2html.vim` vom Vim Kommandomodus konvertiert hervorgehobenen Text nach HTML. Sichern Sie dies mit `:w Datei.html` und `:q`. Dies ist nützlich für C Code, etc.

11.3.8 Aufteilen des Bildschirms mit `vim`

`vim` kann mehrere Dateien in mehrfach aufgeteilten Bildschirmen editieren. Geben Sie `:help usr_08.txt` für Details ein.

Um die Bildschirmdarstellung für verschiedene Dateien aufzuteilen, ist am vi Kommandoprompt folgendes einzugeben:

```
:split andere-Datei
:vsplit andere-Datei
```

Beim Shell-Prompt führt

```
$ vi -o Datei1.txt Datei2.txt   # horizontale Aufteilung
$ vi -O Datei1.txt Datei2.txt   # vertikale Aufteilung
```

zu einem Mehr-Fenster vi.

```
$ vimdiff Datei.txt~ Datei.txt   # überprüfen der Änderungen von
$ vimdiff Datei.en Datei.fr      #  Datei.txt bzw. der Übersetzung
$ gvimdiff Datei.txt~ Datei.txt  # in X
```

ermöglicht einen schönen Blick auf die Unterschiede zwischen der ursprünglichen und der Backup Datei. In SGML erkennt es Tags, so dass der Vergleich von Übersetzungen in diesem Modus sehr gut funktioniert.

Spezielle Cursor Bewegungen mit Strg-W:

```
Strg-W +      vergrößern des Fensters
Strg-W -      verkleinern des Fensters
Strg-W h      wechseln ins linke Fenster
Strg-W j      wechseln in untere Fenster
Strg-W k      wechseln ins obere Fenster
Strg-W l      wechseln ins rechte Fenster
...
```

Nutzen Sie folgendes zur Kontrolle (Synchronisation) des Bildschirm Scrollens:

```
:set scrollbind
:set noscrollbind
```

Kapitel 12

Versions-Kontroll-Systeme

12.1 CVS

Überprüfen Sie `/usr/share/doc/cvs/html-cvsclient`, `/usr/share/doc/cvs/html-info` und `/usr/share/doc/cvsbook` mit `lynx` oder starten Sie `info cvs` und `man cvs` für detaillierte Informationen.

12.1.1 Installation eines CVS Servers

Das folgende Setup erlaubt commit's ins CVS Repository nur durch ein Mitglied der "src" Gruppe und die Administration des CVS nur durch ein Mitglied der "staff" Gruppe. Dies reduziert die Möglichkeit, den Server versehentlich zu misskonfigurieren.

```
# cd /var/lib; umask 002; mkdir cvs # [Woody] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .          # "staff" um bei neuen Projekten mehr
                            # zu restriktieren
# chmod 3775 .              # bei "staff" statt "src", 2775 verwenden
# cvs -d /var/lib/cvs init # es ist sicherer -d anzugeben
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.1.2 Beispiele für CVS Sitzungen

Das folgende stellt die Shell Umgebung für den CVS Repository Zugriff ein.

Anonymes CVS (nur Download)

Entfernter nur lesbarer Zugang:

```
$ export CVSROOT=:pserver:anonymous@cvs.qref.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

Nutzen eines lokalen CVS Servers

Lokaler Zugang von der Shell auf der selben Maschine:

```
$ export CVSROOT=/var/lib/cvs
```

Nutzen eines entfernten CVS pserver

Entfernter Zugriff ohne SSH (verwenden Sie das RSH Protokoll in `cvs`):

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

Dies ist für Lauschangriffe anfällig.

Nutzen von entfernten CVS mittels `ssh`

Entfernter Zugriff mit SSH:

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

oder für SourceForge:

```
$ export CVSROOT=:ext:account@cvs.qref.sf.net:/cvsroot/qref
```

Man kann auch RSA Authentifizierung nutzen (‘Mit weniger Passwörter agieren’ auf Seite 142), was die Passwortabfrage überflüssig macht.

Erzeugen eines neuen CVS Archivs

Für

```
EINTRAG          WERT              BEDEUTUNG
Quellbaum:       ~/Projekt-x       gesamter Quellcode
Projektname:     Projekt-x         der Name dieses Projekts
Vendor Tag:      Hauptzweig        Wert für ganzen Zweig
Release Tag:     Release-initial   Wert für eine spezielle Version
```

Danach

```
$ cd ~/Projekt-x                # wechseln ins Quellverzeichnis
 ... Erzeugen des Quellcodes ...
$ cvs import -m "Starte Projekt-x" Projekt-x Hauptzweig Release-initial
$ cd ..; rm -R ~/Projekt-x
```

Arbeiten mit CVS

Um mit *Projekt-x* unter Verwendung des lokalen CVS Repositories zu arbeiten:

```
$ cd                            # ins Arbeitsverzeichnis wechseln
$ cvs co Projekt-x              # Quellen aus dem CVS kopieren
$ cd Projekt-x
 ... ändern des Codes ...
$ cvs diff -u                   # wie diff -u Repository/ lokal/
$ cvs up -C modifizierte_Datei  # Änderungen rückgängig machen
$ cvs ci -m "Änderungen"        # sichern der lokalen Quellen
$ vi neue_hinzugefügte_Datei    #   ... ins CVS
$ cvs add neue_hinzugefügte_Datei
$ cvs ci -m "Neue Datei neue_hinzugefügte_Datei"
$ cvs up                        # verschmelzen mit der letzten
                                # Version aus dem CVS
 ... zum Erzeugen aller neu erstellten Unterverzeichnisse im CVS
 ... ist "cvs up -d -P" stattdessen zu verwenden
 ... achten Sie auf Ausgaben, die mit "C Dateiname" starten
 ... nicht modifizierter Code wird nach '.#Dateiname.Version'
     verschoben
 ... suchen Sie nach "<<<<<<<" und ">>>>>>>" in der Datei
$ cvs tag Release-1             # hinzufügen des Release Tag
 ... weitere Änderungen ...
$ cvs tag -d Release-1          # entfernen des Release Tag
$ cvs ci -m "mehr Kommentare"
$ cvs tag Release-1             # erneutes hinzufügen des Tag
```

```
$ cd                              # ins Arbeitsverzeichnis wechseln
$ cvs co -r Release-initial -d alt Projekt-x
 ... Originalversion ins alt Verzeichnis kopieren
$ cd alt
$ cvs tag -b Release-initial-bugfixes # erzeuge Zweig (-b)
 ... nun kann mit der alten Version gearbeitet werden (Tag=sticky)
$ cvs update -d -P                # keine leeren Verz. erstellen
 ... Quellbaum hat nun den sticky Tag "Release-initial-bugfixes"
 ... arbeiten in diesem Zweig
$ cvs up -d -P                    # synchronisieren mit von anderen
                             # modifizierten Dateien in diesem Zweig
$ cvs ci -m "einchecken in diesen Zweig"
$ cvs update -kk -A -d -P
 ... entfernen des sticky Tag und Inhalt vergessen
 ... Aktualisierung des Haupt"trunk" ohne Schlüsselwortersetzung
$ cvs update -kk -d -P -j Release-initial-bugfixes
 ... vermengen des Release-initial-bugfixes Zweigs in den
 ... Haupt"trunk" ohne Schlüsselwortersetzung. Beseitigen Sie
 ... Konflikte mit einem Editor.
$ cvs ci -m "vermenge Release-initial-bugfixes"
$ cd
$ tar -cvzf Projekt-x-alt.tar.gz alt # Archiv erstellen, -j für bz2
$ cvs release -d alt              # lokale Quellen entfernen (optional)
```

Nette Optionen zur Erinnerung (als erstes Argument von `cvs` nutzen):

```
-n      Probelauf, hat keinen Effekt
-t      Anzeigen von Mitteilungen zur CVS Aktivität
```

Exportieren von Dateien aus dem CVS

Um die letzte Version aus dem CVS zu nutzen, verwenden Sie “tomorrow” (morgen):

```
$ cvs ex -D tomorrow Modulname
```

Verwalten des CVS

Fügen Sie ein Alias zum Projekt hinzu (lokaler Server):

```
$ su - admin           # ein Mitglied von staff
$ export CVSROOT=/var/lib/cvs
$ cvs co CVSROOT/modules
$ cd CVSROOT
```

```
$ echo "px -a Projekt-x" >>modules
$ cvs ci -m "Nun ist px ein Alias für Projekt-x"
$ cvs release -d .
$ exit                  # oder Strg-D um von su zurückzukehren
$ cvs co -d Projekt px
 ... auschecken von Projekt-x (Alias: px) aus dem CVS ins
 ... Verzeichnis Projekt
$ cd Projekt
 ... Änderungen vornehmen ...
```

12.1.3 Fehlersuche im CVS

Dateirechte im Repository

CVS überschreibt nicht die Datei im Repository, sondern ersetzt sie mit einer anderen. Deshalb sind *Schreibrechte im Repository Verzeichnis* wichtig. Für jedes neue Repository ist folgendes zu starten, um diese Bedingung zu sichern.

```
# cd /var/lib/cvs
# chown -R root:src Repository
# chmod -R ug+rwX   Repository
# chmod 2775 Repository # wenn nötig, auch bei Unterverzeichnissen
```

Das ausführbar-Bit

Das ausführbar-Bit einer Datei wird beibehalten beim Checkout. Wann immer Sie Probleme mit dem ausführbar-Recht in ausgecheckten Dateien haben, ändern Sie die Rechte der Datei im CVS Repository mit dem folgenden Kommando:

```
# chmod ugo-x Dateiname
```

12.1.4 CVS Kommandos

Es folgen CVS Kommandos mit deren Kürzeln.

```
{add|ad|new} [-k kflag] [-m 'Bemerkung'] Dateien...
{admin|adm|rcs} [rcs-Optionen] Dateien...
{annotate|ann} [Optionen] [Dateien...]
{checkout|co|get} [Optionen] Module...
{commit|ci|com} [-lnR] [-m 'Bemerkung fürs Protokoll' | -f Datei] \
        [-r Revision] [Dateien...]
{diff|di|dif} [-kl] [rcsdiff_Optionen] [[-r rev1 | -D Datum1] \
```

```
        [-r rev2 | -D Datum2]] [Dateien...]
{export|ex|exp} [-flNn] -r rev|-D Datum [-d dir] [-k kflag] Modul...
{history|hi|his} [-report] [-flags] [-options args] [Dateien...]
{import|im|imp} [-options] Repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-Optionen [Dateien...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] Module...
{release|re|rel} [-d] Verzeichnisse...
{remove|rm|delete} [-lR] [Dateien...]
{rtag|rt|rfreeze} [-falnR] [-b] [-d] [-r Tag | -D Datum] \
         symbolic_tag Module...
{status|st|stat} [-lR] [-v] [Dateien...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r Tag | -D Datum]  [-f] \
         symbolic_tag [Dateien...]
{update|up|upd} [-AdflPpR] [-d] [-r Tag|-D Datum] Dateien...
```

12.2 Subversion

Subversion ist ein Versions-Kontroll-System der nächsten Generation, das einmal CVS ersetzen soll. Die Entwickler betrachten es zur Zeit als im “alpha” Stadium befindlich, aber es ist wahrscheinlich für die meisten Anwendungen stabil genug. Während dies geschrieben wird, ist Subversion nur in Debian `unstable` verfügbar.

12.2.1 Installation eines Subversion Servers

Das `subversion-server` Meta-Paket hängt von den benötigten Paketen `libapache2-dav-svn` und `subversion-tools` ab, um einen Server aufzusetzen.

Aufsetzen eines Repository

Zur Zeit setzt das `subversion` Paket kein Repository auf, man muss es manuell erzeugen. Eine mögliche Stelle für ein Repository ist in `/var/local/repos`.

Erzeugen des Verzeichnisses

```
# mkdir -p /var/local/repos
```

und der Repository Datenbank:

```
# svnadmin create /var/local/repos
```

Nun ist das Repository für den WWW Server schreibbar zu machen:

```
# chown -R www-data:www-data /var/local/repos
```

Konfiguration von Apache2

Um den Zugriff auf das Repository mittels Nutzerauthentifizierung zu ermöglichen, fügen Sie folgendes zu `/etc/apache2/mods-available/dav_svn.conf` hinzu (oder kommentieren Sie dies aus):

```
<Location /repos>
  DAV svn
  SVNPath /var/local/repos
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /etc/subversion/passwd
  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  <LimitExcept>
</Location>
```

Danach erzeugen Sie eine Nutzerauthentifizierungsdatei mit dem Kommando:

```
htpasswd2 -c /etc/subversion/passwd ein-Nutzer
```

Starten Sie Apache2 neu und Ihr neues Subversion Repository wird unter der URL http://*hostname*/repos verfügbar sein.

12.2.2 Verschieben eines CVS Repositories nach Subversion

12.2.3 Anwendungsbeispiele von Subversion

Die folgenden Abschnitte erklären die Verwendung verschiedener Kommandos in Subversion.

Erzeugen eines neuen Subversion Archivs

Um ein neues Subversion Archiv zu erstellen, verwenden Sie folgendes:

```
$ cd ~/Projekt          # ins Quellcodeverzeichnis wechseln
$ svn import http://localhost/repos Projekt Projektname \
      -m "erster Projektimport"
```

Dies erzeugt ein Verzeichnis namens *Projektname* im Subversion Repository, das die Projektdateien enthält. Schauen Sie unter http://localhost/repos/, um zu sehen, ob die Datei vorhanden ist.

Arbeiten mit Subversion

Arbeiten mit *Projekt-y* und Subversion:

```
$ cd                            # ins Arbeitsverzeichnis wechseln
$ svn co http://localhost/repos/Projekt-y  # Quellcode auschecken
$ cd Projekt-y
 ... Änderungen durchführen ...
$ svn diff                      # wie diff -u Repository/ lokal/
$ svn revert modifizierte_Datei # Änderungen rückgängig machen
$ svn ci -m "Änderungen"        # Änderungen einchecken
$ vi neue_hinzugefügte_Datei
$ svn add neue_hinzugefügte_Datei
$ svn add Verzeichnis1          # alle Dateien unter Verzeichnis1
                                # rekursiv hinzufügen
$ svn add -N Verzeichnis2       # Verz. nicht rekursiv hinzufügen
$ svn ci -m "neue Dateien hinzugefügt"
$ svn up                        # vermenge mit neuester Version
                                # aus Repository
$ svn log                       # zeige alle eingebrachten Änd.
$ svn copy http://localhost/repos/Projekt-y \
      http://localhost/repos/Projekt-y-branch \
      -m "erzeuge Zweig von Projekt-y"
$ svn copy http://localhost/repos/Projekt-y \
      http://localhost/repos/Proj-y_release1.0 \
      -m "Projekt-y 1.0 Release"# fügte release Tag hinzu
 ... Es ist zu beachten, dass das Erzeugen eines Zweigs und eines
 ... Tags das selbe ist. Der einzige Unterschied ist, dass Zweige
 ... "committed" werden, Tags nicht.
 ... Änderungen im Zweig durchführen ...

$ # vermengen der Kopie des Zweiges mit Hauptkopie
$ svn merge http://localhost/repos/Projekt-y \
   http://localhost/repos/Projekt-y-branch
$ svn co -r 4 http://localhost/repos/Projekt-y # Rev. 4 besorgen
```

Kapitel 13

Programmierung

Verwenden Sie nicht “test” als Namen für eine ausführbare Testdatei. `test` ist ein Shell builtin.

13.1 Wo man startet

Referenzen:

- Dokumente und Beispiele unter `/usr/share/doc/`*`Paket`*
- Unix / Programming Information (`http://arioch.unomaha.edu/~jclark/#info`)
- *Linux Programming Bible* (John Goerzen/IDG books)

Viele lange info Dokumente können als Taschenbuch von GNU (`http://www.gnu.org/`) erhalten werden.

Die nächsten vier Abschnitte enthalten Beispielskripte in verschiedenen Sprachen, die eine Textdatei mit Zugangsinformationen erzeugen, welche an `/etc/passwd` mittels einer Stapelverarbeitung wie das `newusers` Programm angehangen werden. Jedes Skript erfordert als Eingabe eine Datei mit Zeilen der Gestalt `Vorname Nachname Passwort`. (Homeverzeichnisse werden mit diesen Skripten nicht erzeugt.)

13.2 Shell

Das Lesen von Shellskripten ist der **beste** Weg, um zu verstehen, wie ein Unix-artiges System arbeitet. Ich gebe hier einige Hinweise zur Shellprogrammierung an. Lesen Sie Shell-Fehler (`http://www.greenend.org.uk/rjk/2001/04/shell.html`) um aus Fehlern zu lernen.

13.2.1 Bash – interaktive GNU Standard Shell

Referenzen für Bash:

- `bash(1)`
- `info bash`
- das LDP BASH Programming - Introduction HOWTO (`http://www.tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html`) als Information für Starter.
- `mc /usr/share/doc/bash/examples/ /usr/share/doc/bash/` (Installieren Sie das `bash-doc` Paket für Beispieldateien.)
- *Learning the bash Shell*, 2. Ausgabe (O'Reilly)

Kurzes Programmbeispiel (erzeugt Zugangswerte für `newusers` von der Standardeingabe):

```
#!/bin/bash
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.2.2 POSIX Shells

Verschiedene Pakete bieten eine POSIX Shell in Debian an:

- `dash` (Sarge)
  - Priority: optional
  - Installed-Size: 176
  - Kleinste und schnellste – das Beste zum Booten
- `ash` (Woody)
  - Priority: optional
  - Installed-Size: 180
  - Kleiner und viel schneller – gut für Bootprozess
- `bash`
  - Essential: yes
  - Priority: required
  - Installed-Size: 580
  - Größer und vielfältig – viele implementierte Erweiterungen
- `pdksh`
  - Priority: optional
  - Installed-Size: 408
  - Komplett AT&T ksh ähnlich

Wollen Sie portable Shellskripte schreiben, ist es am besten es als POSIX Shellskript zu schreiben. Verwenden Sie `/bin/sh` verlinkt auf `ash` (oder `dash`), um dessen POSIX Konformität zu testen. Vermeiden Sie Skripte mit **bashism** oder **zshism** zu schreiben, was der `csh` Syntax ähnelt. Vermeiden Sie zum Beispiel:

- `if [ foo == bar ] ; then ...`

- `diff -u Datei.c{.orig,}`
- `mkdir /foo{bar,baz}`

13.2.3 Shellparameter

Verschiedene **spezielle Parameter** zum Einprägen:

```
$0      = Name der Shell oder des Shellskripts
$1      = erstes Shellargument
 ...
$9      = neuntes Shellargument
$#      = Anzahl der Parameter
"$*"    = "$1 $2 $3 $4 ... $n"
"$@"    = "$1" "$2" "$3" "$4" ... "$n"
$?      = Exit-Status des zuletzt ausgeführten Kommandos
$$      = Prozessnummer (PID) dieses Shellskripts
$!      = PID des zuletzt ausgeführten Hintergrundkommandos
```

Grundlegende **Parameterauswertungen** zum Einprägen:

Ausdruck	Wenn *var* gesetzt ist	Wenn *var* nicht gesetzt ist
${*var*:-*string*}	$*var*	*string*
${*var*:+*string*}	*string*	null
${*var*:=*string*}	$*var*	*string* (und setzt *var*=*string*)
${*var*:?*string*}	$*var*	(*string* Ausgabe und Abbruch)

Der Doppelpunkt ':' in all diesen Operatoren ist optional.

- Mit ':' = Operatortest für "existiert" und "nicht Null".
- Ohne ':' = Operatortest nur für "existiert".

Grundlegende **Parametersubstitutionen** zum Einprägen:

Ausdruck	Ergebnis
${*var%suffix*}	Entferne kleinstes *suffix* Muster
${*var%%suffix*}	Entferne größtes *suffix* Muster
${*var#präfix*}	Entferne kleinstes *präfix* Muster
${*var##präfix*}	Entferne größtes *präfix* Muster

13.2.4 Shellumleitung

Grundlegende **Umleitungen** zum Einprägen ([*n*] ist eine optionale Nummer):

[*n*]> *Datei*	Umleiten der Standardausgabe (oder *n*) zu *Datei*.
[*n*]>> *Datei*	Anhängen der Standardausgabe (oder *n*) zu *Datei*.
[*n*]< *Datei*	Umleiten der Standardeingabe (oder *n*) von *Datei*.
[*n1*]>&*n2*	Umleiten der Standardausgabe (oder *n1*) zu *n2*.
2> *Datei* >&2	Umleiten der Standard- und Fehlerausgabe zu *Datei*.
| *Kommando*	Standardausgabe an *Kommando* weiterreichen.
2>&1| *Kommando*	Standard- und Fehlerausg. an *Kommando* weiterreichen.

13.2.5 Bedingte Ausdrucke in der Shell

Jedes Kommando gibt einen **Exit-Status** zurück, was für einen bedingten Ausdruck verwendet werden kann:

- Erfolg: 0 (wahr)
- Fehler: 1–255 (falsch)

Es ist zu beachten, dass die Verwendung des 0 Werts im Sinne von "wahr" sich von den üblichen Konventionen in anderen Bereichen der Programmierung unterscheidet. Auch ist '[' das Äquivalent des `test` Kommandos, das sein Argument bis zu ']' als bedingten Ausdruck auswertet.

Grundlegende **bedingte Ausdrucke** zum Einprägen sind:

```
Kommando && bei_Erfolg_dies_starten || true
Kommando || bei_Misserfolg_dies_starten

if [ bedingter_Ausdruck ]; then
 bei_Erfolg_dies_starten
else
 bei_Misserfolg_dies_starten
fi
```

Es wurde `|| true` verwendet, um sicherzustellen, dass das Shellskript in dieser Zeile nicht versehentlich abgebrochen wird, wenn die Shell (oder `set`) mit der Option `-e` gestartet wird.

Dateivergleichsoperatoren im bedingten Ausdruck sind:

```
Ausdruck          Wahr wenn ...
-e Datei          Datei existiert.
-d Datei          Datei existiert und ein Verzeichnis ist.
-f Datei          Datei existiert und eine reguläre Datei ist.
-w Datei          Datei existiert und schreibbar ist.
-x Datei          Datei existiert und ausführbar ist.
Datei1 -nt Datei2 Datei1 neuer als Datei2 ist. (Modifizierungszeit)
Datei1 -ot Datei2 Datei1 älter als Datei2 ist. (Modifizierungszeit)
Datei1 -ef Datei2 beide die selbe Device- und Inode-Nummer haben.
```

Stringvergleichsoperatoren im bedingten Ausdruck sind:

```
Ausdruck       Wahr wenn ...
     -z str    die Länge von str Null ist.
     -n str    die Länge von str nicht Null ist.
str1 == str2   str1 und str2 gleich sind.
str1 =  str2   str1 und str2 gleich sind.
   ("=" sollte statt "==" für POSIX Konformität verwendet werden)
str1 != str2   str1 und str2 ungleich sind.
str1 <  str2   str1 vor str2 einsortiert wird (locale abhängig).
str1 >  str2   str1 nach str2 einsortiert wird (locale abhängig).
```

Arithmetische Ganzzahlvergleiche im bedingten Ausdruck sind `-eq`, `-ne`, `-lt`, `-le`, `-gt` und `-ge`.

13.2.6 Kommandozeilenbearbeitung

Die Shell führt ein Skript wie folgt aus:

- aufsplitten in **Token** durch Metazeichen: LEERZEICHEN, TABULATOR, NEUEZEILE, ;, (,), <, >, |, &
- überprüfen auf **Schlüsselworte** außerhalb von "..." und '...' (Schleife)
- expandieren der **Aliase** außerhalb von "..." und '...' (Schleife)
- expandieren der **geschweiften Klammern**, `a{1,2}` -> `a1 a2` außerhalb von "..." und '...'
- expandieren der **Tilde**, *~Nutzer* -> *Nutzer*'s Homeverzeichnis außerhalb von "..." und '...'
- expandieren der **Parameter**, $*PARAMETER* außerhalb von '...'
- expandieren der **Kommandosubstitutionen**, $(*Kommando*) außerhalb von '...'
- aufsplitten in **Wörter** mit $IFS außerhalb von "..." und '...'
- expandieren von *?[] in **Pfadnamen** außerhalb von "..." und '...'
- schauen nach **Kommando**
  - Funktion
  - builtin
  - Datei in $PATH
- Schleife

Einfache Anführungszeichen innerhalb von doppelten Anführungszeichen haben keinen Effekt.

13.3 Awk

Referenzen für Awk:

- *Effective awk Programming*, 3. Ausgabe (O'Reilly)
- *Sed & awk*, 2. Ausgabe (O'Reilly)
- `mawk(1)` und `gawk(1)`
- `info gawk`

Kurzes Programmbeispiel (erzeugt `newusers` Kommandoeintrag):

```
#!/usr/bin/awk -f
# Skript zum Erzeugen einer für das 'newusers' Kommando geeigneten
# Datei aus einer Datei bestehend aus Nutzer IDs und Passwörtern
# in der Form:  Vorname Nachname Passwort
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Verteilbar unter GNU GPL v 2 oder (je nach Ihrer Wahl) einer
# späteren Version.
```

```
# Dieses Programm wird OHNE JEGLICHE HAFTUNG vertrieben.

BEGIN {
    # zuweisen der anfänglichen UID, GID
    if ( ARGC > 2 ) {
        startuid = ARGV[1]
        delete ARGV[1]
    }
    else {
        printf( "Verwendung: newusers startUID Datei\n" \
        " wobei:\n" \
        "  startUID die erste Nutzer ID zum Hinzufügen ist und\n"\
        "  Datei eine Eingabedatei der folgenden Gestalt ist:\n" \
        "    Vorname Nachname Passwort\n" \
        )
        exit
    }

    infile = ARGV[1]
    printf( "Erste UID: %s\n\n", startuid )
}

/^#/ { next }

{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
        user, passwd, uid, gid, first, last, user \
        )
}
```

Zwei Pakete bieten POSIX awk in Debian an:

- mawk
  - Priority: required
  - Installed-Size: 228
  - Kleiner und viel schneller – gut zur Standardinstallation
  - Compilezeit Limitierungen bestehen
    - NF = 32767
    - sprintf buffer = 1020
- gawk

  - Priority: optional
  - Installed-Size: 1708
  - Größer und reichhaltig – viele implementierte Erweiterungen
    * System V Release 4 Version von UNIX
    * Bell Labs awk
    * GNU spezifisch

13.4 Perl

Dies ist **der** Interpreter auf einem Unix-artigen System.

Referenzen für Perl:
- `perl(1)`
- *Programming Perl*, 3. Ausgabe (O'Reilly)

Kurzes Programmbeispiel (erzeugt `newusers` Kommandoeintrag):

```
#!/usr/bin/perl
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
        if (/^#/) { next;}
        chop;
        $pid++;
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                  ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Installieren eines Perl Moduls *Modul*:

```
# perl -MCPAN -e 'install Modul'
```

13.5 Python

Dies ist ein netter objektorientierter Interpreter.

Referenzen für Python:
- `python(1)`
- *Learning Python* (O'Reilly).

Kurzes Programmbeispiel (erzeugt `newusers` Kommandoeintrag):

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Portiert von awk Skript durch KMSelf Sat Aug 25 20:47:38 PDT 2001
# Dieses Programm wird OHNE JEGLICHE HAFTUNG vertrieben.

def usages():
    print \
"Verwendung: ", sys.argv[0], " start_UID [Dateiname]\n" \
"\tstartUID ist die erste Nutzer ID zum Hinzufügen.\n" \
"\tDateiname ist eine Eingabedatei. " \
"Ohne Angabe wird die Standardeingabe verwendet.\n\n" \
"Format der Eingabedatei:\n" \
"\tVorname Nachname Passwort\n"
                return 1

def parsefile(startuid):
    #
    # filtern
    #
    uid = startuid
    while 1:
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # obiges stürzt bei falscher Parameteranzahl ab :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" %  \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID Start von: %d\n" % uid
        if len(sys.argv) > 1:
            infilename   = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
```

```
        #print "# Lese Datei von: %s\n\n" % infilename
    else:
        infile = sys.stdin
    parsefile(uid)
```

13.6 Make

Referenzen für Make:

- `info make`
- `make(1)`
- *Managing Projects with make*, 2. Ausgabe (O'Reilly)

Einfache automatische Variablen:

Regelsyntax:

```
Ziel: [ Voraussetzungen ... ]
 [TAB]  Kommando1
 [TAB]  -Kommando2 # ignoriere Fehler
 [TAB]  @Kommando3 # unterdrücke Kommandozeilenausgabe
```

Hier ist `[TAB]` ein Tabulator. Jede Zeile wird von der Shell nach einer Variablensubstitution interpretiert. Verwenden Sie \ am Ende einer Zeile zur Fortsetzung des Skripts. Statt $ ist $$ für Umgebungsvariablen zu schreiben.

Implizite Regeln für *Ziel* und *Voraussetzungen* können beispielsweise wie folgt geschrieben werden:

```
%: %.c header.h
```

oder

```
%.o: %.c header.h
```

Hier enthält *Ziel* das Zeichen % (exakt einmal). % passt auf jeden nicht leeren Teilstring in den aktuellen Zieldateinamen. *Voraussetzungen* verwendet % ähnlich um auszudrücken, wie sich diese Namen zum aktuellen Ziel verhalten.

Suffixregeln sind der **veraltete** Weg zur Definition impliziter Regeln für `make`. Sie werden noch in GNU `make` zur Kompatibilität unterstützt, aber man sollte äquivalente Musterregeln wann immer möglich verwenden:

```
alte Suffixregel --> neue Musterregel
.c:              --> %  : %.c
.c.o:            --> %.o: %.c
```

Automatische Variablen für diese Regel:

```
foo.o: neu1.c neu2.c alt1.c neu3.c
$@ == foo.o                          (Ziel)
$< == neu1.c                         (erstes der neueren Objekte)
$? == neu1.c neu2.c neu3.c           (alle neueren Objekte)
$^ == neu1.c neu2.c alt1.c neu3.c    (alle)
$* == '%' aktuelles Ziel ohne Suffix.
```

Variablenreferenzen:

```
foo1 := bar    # einmalige Expandierung
foo2  = bar    # rekursive Expandierung
foo3 += bar    # anhängen
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Extracts "directory"
 $(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Starten Sie `make -p -f/dev/null`, um alle internen automatischen Regeln zu sehen.

13.7 C

Vorbereitung:

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc
```

Referenzen für C:
- `info libc` (Referenz der C Bibliotheksfunktionen)
- `gcc(1)`
- `jede_C_Bibliotheksfunktion(3)`
- Kernighan & Ritchie, *Die Programmiersprache C*, 2. Ausgabe (Prentice Hall).

13.7.1 Ein einfaches C Programm (`gcc`)

Ein einfaches Beispiel zum Kompilieren von `example.c` mit einer Bibliothek `libm` in eine ausführbare Datei namens `Testbeispiel`:

```
$ cat > example.c <<EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* Pufferüberlauf verhindern */
        y[10] = '\0'; /* füllen, so dass String mit '\0' endet */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF

$ gcc -Wall -g -o Testbeispiel example.c -lm
$ ./Testbeispiel
        1, 2.915, ./Testbeis,     (null)
$ ./Testbeispiel 1234567890qwertz
        2, 3.082, ./Testbeis, 1234567890qwertz
```

Hier wird `-lm` benötigt, um mit der Bibliothek `libm` für `sqrt()` zu linken. Die eigentliche Bibliothek ist in `/lib/` mit dem Dateinamen `libm.so.6`, was ein symbolischer Link auf `libm-2.1.3.so` ist.

Schauen Sie auf den letzten Parameter im Ausgabetext. Es gibt mehr als 10 Buchstaben, obwohl `%10s` spezifiziert ist.

Die Verwendung von Funktionen die Zeiger auf Speicherbereiche ohne Bereichscheck nutzen, wie `sprintf` und `strcpy` wird missbilligt, um das Ausnutzen von Pufferüberläufen zu verhindern, die obige Überlaufeffekte verwenden. Stattdessen sollte man `snprintf` und `strncpy` verwenden.

13.7.2 Fehlersuche

Fehlersuche (Debugging) mit `gdb`

Vorbereitung:

```
# apt-get install gdb
```

Referenzen für `gdb`:

- `info gdb` (Tutorial)
- `gdb(1)`

- http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html

Verwenden Sie gdb, um ein mit der -g Option kompiliertes Programm zu debuggen. Viele Kommandos können abgekürzt werden. Vervollständigungen arbeiten wie in der Shell mit der Tabulator Taste.

```
$ gdb Programm
(gdb) b 1                # Haltepunkt in Zeile 1 setzen
(gdb) run arg1 arg2 arg3 # Programm starten
(gdb) next               # nächste Zeile
...
(gdb) step               # einen Schritt vorwärts
...
(gdb) p parm             # parm ausgeben
...
(gdb) p parm=12          # Wert auf 12 setzen
```

Um aus Emacs heraus zu debuggen, wird auf 'Zusammenfassung der Editorkommandos (Emacs, Vim)' auf Seite 163 verwiesen.

Überprüfen der Abhängigkeiten von Bibliotheken

Verwenden Sie ldd, um die Abhängigkeiten eines Programms von Bibliotheken zu bestimmen:

```
$ ldd /bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Damit ls in einer chroot Umgebung funktioniert, müssen die obigen Bibliotheken in der chroot Umgebung vorhanden sein.

Die folgenden Kommandos sind auch nützlich:
- strace: verfolgt Systemaufrufe und Signale
- ltrace: verfolgt Bibliotheksaufrufe

Debuggen mit Tools zur Erkennung von Speicherlecks

Es gibt verschiedene Tools zur Erkennung von Speicherlecks in Debian.
- njamd
- valgrind
- dmalloc
- electric-fence

- memprof
- memwatch (nicht paketiert, verfügbar unter GNU memwatch (http://www.gnu.org/directory/devel/debug/memwatch.html).)
- mpatrol
- leaktracer
- libgc6
- Insure++ von Parasoft (http://www.parasoft.com). (nicht frei, kommerziell)

Überprüfen Sie auch Debugging Tools für dynamische Speicherzuordnung und -Management (http://www.cs.colorado.edu/homes/zorn/public_html/MallocDebug.html).

13.7.3 Flex – ein besseres Lex

flex ist ein schneller lexikalischer Analysegenerator.

Referenzen für flex:
- info flex (Tutorial)
- flex(1)

Sie müssen Ihre eigenen Funktionen main() und yywrap() implementieren oder Ihr Programm.l sollte wie folgt aussehen, um ohne eine Bibliothek zu kompilieren (yywrap ist ein Makro; %option main aktiviert %option noyywrap implizit):

```
%option main
%%
.|\n    ECHO ;
%%
```

Alternativ kann mit der -lfl Linker Option am Ende Ihrer cc Kommandozeile kompiliert werden (wie AT&T-Lex mit -ll). Keine %option wird in diesem Fall benötigt.

13.7.4 Bison – ein besseres Yacc

Einige Pakete bieten einen Yacc kompatiblen LALR Parser Generator in Debian an:
- bison: GNU LALR Parser Generator
- byacc: Der Berkeley LALR Parser Generator
- byyacc: Rückfolgender Parser Generator basierend auf byacc

Referenzen für bison:
- info bison (Tutorial)
- bison(1)

Sie müssen Ihre eigenen main() und yyerror() Funktionen implementieren. main() ruft yyparse(), was wiederum yylex() aufruft, das gewöhnlich von FleX erzeugt wird.

```
%%

%%
```

13.7.5 Autoconf

`autoconf` ist ein Tool zum Erzeugen von Shellskripten, die Softwarequellpakete automatisch so konfigurieren, dass sie sich an viele UNIX-artige Systeme unter Verwendung des vollständigen GNU Build Systems anpassen.

`autoconf` erzeugt das Konfigurationsskript `configure`. `configure` erzeugt automatisch ein angepasstes `Makefile` aus `Makefile.in`.

Kompilieren und Installieren eines Programms

Debian ändert keine Dateien unter `/usr/local` (vergleichen Sie 'Unterstützung von Vielfalten' auf Seite 24). Kompilieren Sie ein Programm aus den Quellen, so sollten Sie es in `/usr/local` installieren, damit es nicht mit Debian interferiert.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make
$ make install # dies kopiert die Dateien ins System
```

Deinstallation eines Programms

Wenn Sie noch den Quellcode haben, dieser `autoconf/automake` nutzt und Sie noch wissen, wie Sie es konfiguriert haben, verfahren Sie wie folgt:

```
$ ./configure alle-verwendeten-Optionen
# make uninstall
```

Sind Sie sich absolut sicher, dass der Installationsprozess Dateien nur unter `/usr/local` ablegt und es nichts wichtiges darunter gibt, können Sie alles wie folgt löschen:

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Sind Sie nicht sicher, wo Dateien installiert sind, sollten Sie einen Blick auf `checkinstall` werfen, was einen leeren Pfad bei der Deinstallation liefert.

13.8 Dokument Aufbereitung

13.8.1 Das `roff` Satzsystem

Traditionell ist `roff` das wichtigste Unix Textbearbeitungssystem.

Vergleichen Sie `roff(7)`, `groff(7)`, `groff(1)`, `grotty(1)`, `troff(1)`, `groff_mdoc(7)`, `groff_man(7)`, `groff_ms(7)`, `groff_me(7)`, `groff_mm(7)` und `info groff`.

Es gibt eine gute Einführung zu `-me` Makros. Haben Sie `groff` (1.18 oder neuer), suchen Sie `/usr/share/doc/groff/meintro.me.gz` und führen Sie das folgende aus:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
     groff -Tascii -me - | less -R
```

Das folgende erzeugt eine einfache Textdatei:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
    GROFF_NO_SGR=1 groff -Tascii -me - | col -b -x > meintro.txt
```

Verwenden Sie die PostScript Ausgabe zum Drucken.

```
$ groff -Tps meintro.txt | lpr
$ groff -Tps meintro.txt | mpage -2 | lpr
```

13.8.2 SGML

Vorbereitung:

```
# apt-get install debiandoc-sgml debiandoc-sgml-doc
```

Referenzen für `debiandoc-sgml`:

- `/usr/share/doc/debiandoc-sgml-doc`
- `debiandoc-sgml(1)`
- *DocBook: The Definitive Guide* (`/usr/share/doc/docbook-defguide/html/docbook.html`) von Walsh und Muellner (O'Reilly) (Paket `docbook-defguide`)

SGML ermöglicht das Verwalten mehrerer Dokumentformate. Ein einfaches SGML System ist Debiandoc, was hier verwendet wird. Dies erfordert kleinere Umwandlungen der Textdateien für die folgenden Buchstaben:

- "<" –> `<`
- ">" –> `>`
- " " –> ` ` (nicht trennbares Leerzeichen)
- "&" –> `&`
- "%" –> `%`
- "©" –> `©`
- "–" –> `–`
- "—" –> `—`

Um einen Abschnitt als nicht ausgebbaren Kommentar zu markieren, wird folgendes verwendet:

```
<!-- Darlegen des Sachverhalts ... -->
```

Für einen Abschnitt mit abschaltbarem Kommentar:

```
<![ %FIXME; [ Darlegen des Sachverhalts ... ]]>
```

In SGML gewinnt die *erste Definition* einer Entität. Zum Beispiel:

```
<!entity % qref "INCLUDE">
<![ %qref; [ <!entity param "Daten 1"> ]]>
<!entity param "Daten 2">
&param;
```

Dies endet mit "Daten 1". Enthält die erste Zeile "IGNORE" anstatt "INCLUDE", endet es mit "Daten 2" (die zweite Zeile ist eine bedingte Anweisung). Es ist auch möglich, einzelne wiederholende Textabschnitte separat vom Inhalt zu definieren.

```
<!entity wessen "mein">
Hallo &wessen; Freund.
Dies ist &wessen; Buch.
```

Dies ergibt das folgende:

```
Hallo mein Freund.
Dies ist mein Buch.
```

Vergleichen Sie mit dem kurzen SGML Beispiel `sample.sgml` im Beispielverzeichnis (`examples/`).

Wenn SGML Dokumente größer werden, hat TeX, das intern zur Textbearbeitung verwendet wird, manchmal Probleme. Vergleichen Sie 'TeX/LaTeX' auf dieser Seite.

13.8.3 TeX/LaTeX

Vorbereitung:

```
# tasksel # wählen Sie Miscellaneous  --> TeX/LaTeX-Umgebung
```

Referenzen für LaTeX:

- The teTeX HOWTO: The Linux-teTeX Local Guide (`http://www.tldp.org/HOWTO/TeTeX-HOWTO.html`)
- `tex(1)`

- `latex(1)`
- *The TeXbook* von Donald E. Knuth (Addison-Wesley) [1]
- *Das LaTeX-Handbuch* von Leslie Lamport (Addison-Wesley)
- *Der LaTeX-Begleiter* von Goossens, Mittelbach, Samarin (Addison-Wesley)

Dies ist das mächtigste Satztextprogramm. Viele SGML Bearbeitungsprogramme verwenden dies zur Textbearbeitung im Hintergrund. Lyx, dass mit den Paketen `lyx`, `lyx-xforms` oder `lyx-qt` bereitgestellt wird, bietet eine nette WYSIWYG Umgebung zum Editieren für LaTeX, zu der viele Emacs und Vim als Quellcode-Editor wählen.

Es gibt viele Online-Verweise:

- teTeX - A Documentation Guide (`/usr/share/doc/texmf/newhelpindex.html`) (`tetex-doc` Paket)
- LaTeX2e-Kurzbeschreibung (`/usr/share/doc/texmf/latex/general/l2kurz.dvi.gz`)
- Viele weitere deutschsprachige Dokumente zu LaTeX (`ftp://ftp.dante.de/tex-archive/info/german/`) (u.a. *Das kleine TeX-Buch*, eine deutscher Auszug aus *The TeXbook*)
- A Quick Introduction to LaTeX (`http://www.msu.edu/user/pfaffben/writings/`)
- A Simple Guide to Latex/Lyx (`http://www.stat.rice.edu/~helpdesk/howto/lyxguide.html`)
- Word Processing Using LaTeX (`http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_basic/latex_basic.html`)
- Local User Guide to teTeX/LaTeX (`http://supportweb.cs.bham.ac.uk/documentation/LaTeX/lguide/local-guide/local-guide.html`)

Wenn Dokumente größer werden, hat TeX manchmal Probleme. Um dies zu korrigieren, muss die Pool-Größe in `/etc/texmf/texmf.cnf` erhöht werden (oder editieren Sie besser `/etc/texmf/texmf.d/95NonPath` und starten Sie `update-texmf`).

13.8.4 Literate Programming

Anstatt Code zu schreiben, der Dokumentationen enthält, schreibt der gebildete Programmierer (literate programmer) Dokumentation die Code enthält. Dieser Ansatz sichert eine gute Dokumentation für ein Programm.

Für weitere Informationen zu literate-programming wird auf Literate Programming (`http://www.literateprogramming.com/`) verwiesen.

[1] Der TeX-Quellcode dieses Buches ist unter `ftp://ftp.dante.de/pub/tex/systems/knuth/tex/texbook.tex` erhältlich. `ftp://ftp.dante.de/pub/tex/systems/knuth/lib/manmac.tex` enthält die meisten der benötigten Makros. Das Dokument kann mit `tex` bearbeitet werden, nachdem die Zeilen 7 bis 10 auskommentiert und `\input manmac \proofmodefalse` hinzugefügt wurde. Es ist sehr empfehlenswert dieses Buch (und alle anderen Bücher von Donald E. Knuth) zu kaufen, statt die Online-Version zu verwenden, aber der Quellcode ist ein sehr gutes Beispiel für TeX-Code!

Noweb

Vorbereitung:

```
# apt-get install nowebm
```

Referenzen für Noweb:

- Noweb — A Simple, Extensible Tool for Literate Programming (http://www.eecs.harvard.edu/~nr/noweb/)
- noweb(1)

Dies ist ein WEB-artiges literate-programming Werkzeug, das einfacher ist, sowie erweiterbar und sprachunabhängig. [2] Wenn noweb gestartet wird, schreibt es den Programmquellcode in Ausgabedateien, die in der noweb Datei angegeben sind, und es erstellt eine TeX-Datei für die Dokumentation.

Das Debian Paket ifupdown ist ein gutes Beispiel.

```
$ apt-get source ifupdown
$ cd ifupdown*
$ make ifupdown.pdf ifupdown.ps
```

Doxygen

Vorbereitung:

```
# apt-get install doxygen doxygen-doc doxygen-gui
```

Referenzen für Doxygen (von doxygen erstellt!):

- Homepage (http://www.doxygen.org/)
- /usr/share/doc/doxygen-doc/html/index.html

Es kann HTML, RTF, Unix Manual-Pages, PostScript und PDF (durch LaTeX) Dokumentation für C++, C, Java, IDL und zum Teil PHP sowie C# Programme erstellen. Doxygen ist kompatibel zu JavaDoc (1.1), Qt-Doc, KDOC und wurde speziell entworfen, um Projekte die Troll Tech's Qt (http://www.trolltech.no/qt/) Bibliothek nutzen, zu unterstützen. Es erstellt Include-Abhängigkeitsdiagramme, Diagramme zu Zusammengehörigkeiten und grafische Darstellungen der Klassenhierarchien sogar für nicht dokumentierte Programme. Die Ausgabe ist ähnlich zu Qt's Dokumentation.

[2] Dieses WEB hat **nichts** zu tun mit dem World Wide Web. WEB (für PASCAL) und CWEB (für C/C++) sind traditionelle literate-programming Werkzeuge.

13.9 Paketerzeugung

Vorbereitung:

```
# apt-get install debian-policy developers-reference \
                  maint-guide dh-make debhelper
# apt-get install packaging-manual # für Potato
```

Referenzen für die Paketerzeugung:

- 'Das Debian Paketverwaltungssystem' auf Seite 11 (Grundlagen)
- Debian New Maintainers' Guide (Tutorial)
- `dh-make(1)`
- Debian Developer's Reference (sehr praxisnah)
- Debian Policy Manual (die ultimative Dokumentation)
- Packaging Manual (Potato)

13.9.1 Paketerzeugung für ein einzelnes Programm

Eine schnelle und unsaubere Methode der Paketerzeugung für ein einzelnes Programm ist nach Joey Hess:

```
# mkdir -p mypkg/usr/bin mypkg/DEBIAN
# cp binary mypkg/usr/bin
# cat > mypkg/DEBIAN/control
Package: mypackage
Version: 1
Architecture: i386
Maintainer: Joey Hess <joeyh@debian.org>
Description: my little package
 Don't expect much.
^D
# dpkg-deb -b mypkg
```

13.9.2 Paketerzeugung mit Tools

Verwenden Sie `dh_make` aus dem `dh-make` Paket, um eine solide Grundlage für ein Paket zu gewinnen. Folgen Sie danach den Anweisungen in `dh-make(1)`. Dies nutzt `debhelper` in `debian/rules`.

Ein älterer Zugang ist das Verwenden von `deb-make` aus dem `debmake` Paket. Dies nutzt keine `debhelper` Skripte und hängt nur von der Shell ab.

Für Beispiele von Paketen mit mehreren Quelldateien vergleichen Sie mit "mc" (`dpkg-source -x mc_4.5.54.dsc`), das "sys-build.mk" von Adam Heath (`<doogie@debian.org>`) nutzt und "glibc" (`dpkg-source -x glibc_2.2.4-1.dsc`), das ein anderes System von Joel Klecker (`<espy@debian.org>`) nutzt.

Kapitel 14

GnuPG

Referenzen:
- `gpg(1)`.
- `/usr/share/doc/gnupg/README.gz`
- *GNU privacy handbook* in `/usr/share/doc/gnupg-doc/GNU_Privacy_Handbook/` (installieren Sie das `gnupg-doc` Paket)

14.1 Installation von GnuPG

```
# gpg --gen-key                  # erzeugen eines neuen Schlüssels
# gpg --gen-revoke Nutzer_ID     # erzeugen eines Widerrufungs-
                                 # schlüssels für Nutzer_ID
# host -l pgp.net |grep www |less# pgp Schlüsselserver bestimmen
```

Zum aktuellen Zeitpunkt sind gute Schlüsselserver:

```
keyserver wwwkeys.eu.pgp.net
keyserver wwwkeys.pgp.net
```

Hier muss man vorsichtig sein **nicht mehr als 2 Unterschlüssel zu erzeugen**. Sollten Sie dies tun, werden die Schlüsselserver auf pgp.net Ihren Schlüssel **korrumpieren**. Nutzen Sie das neuere `gnupg` (>1.2.1-2), um mit diesen korrupten Unterschlüssel umzugehen. Vergleichen Sie `http://fortytwo.ch/gpg/subkeys`.

Es kann auch nur ein Schlüsselserver in `$HOME/.gnupg/options` angegeben werden.

Unglücklicherweise funktioniert das folgende nicht mehr:

```
keyserver search.keyserver.net
keyserver pgp.ai.mit.edu
```

14.2 Verwendung von GnuPG

Handhabung von Dateien:

```
$ gpg [Optionen] Kommando [args]
$ gpg {--armor|-a} {--sign|-s} Datei # signiere Datei in Datei.asc
$ gpg --clearsign Datei              # clear-sign Nachricht
$ gpg --clearsign --not-dash-escaped Patchdatei # clear-sign
                                     # Patchdatei
$ gpg --verify Datei                 # überprüfe clear-signed Datei
$ gpg -o Datei.sig {-b|--detach-sig} Datei # erzeuge losgelöste
                                     # Signatur
$ gpg --verify Sign Datei            # überprüfe Datei mit Sign
$ gpg -o crypt_Datei {--recipient|-r} name {--encrypt|-e} Datei
                              # Public-Key Verschlüsselung für name
$ gpg -o crypt_Datei {--symmetric|-c} Datei # sym. Verschlüsselung
$ gpg -o Datei --decrypt crypt_Datei # Entschlüsselung
```

14.3 Umgang mit GnuPG

Schlüsselmanagement:

```
$ gpg --edit-key Nutzer_ID   # "help" für Hilfe, interaktiv
$ gpg -o Datei --exports     # exportiere alle Schlüssel nach Datei
$ gpg --imports Datei        # importiere alle Schlüssel von Datei
$ gpg --send-keys Nutzer_ID  # sende Schlüssel von Nutzer_ID zum
                             # Schlüsselserver
$ gpg --recv-keys Nutzer_ID  # empfange Schlüssel von Nutzer_ID
                             # vom Schlüsselserver
$ gpg --list-keys Nutzer_ID  # liste Schlüssel von Nutzer_ID auf
$ gpg --list-sigs Nutzer_ID  # liste Signatur von Nutzer_ID auf
$ gpg --check-sigs Nutzer_ID # überprüfe Signatur von Nutzer_ID
$ gpg --fingerprint Nutzer_ID# überprüfe Fingerabdruck von Nutzer_ID
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys
                             # unbekannte Schlüssel besorgen
  # aktualisiere Schlüssel für alle unbekannten Signaturen.
```

Vertrauenscode:

```
-         Kein Besitzervertrauen zugewiesen / noch nicht berechnet.
e         Vertrauensberechnung schlug fehl.
q         Nicht genug Informationen für Berechnung.
```

```
n          Niemals diesem Schlüssel vertrauen.
m          Gerade noch vertrauenswert.
f          Voll vertrauenswert.
u          Äußerst vertrauenswert.
```

Das folgende lädt meinen Schlüssel "*A8061F32*" auf mehrere Schlüsselserver:

```
$ for xx in us es cz de dk uk ch net.uk earth.net.uk; \
$ do gpg --keyserver wwwkeys.$xx.pgp.net --send-keys A8061F32; done
```

14.4 Benutzung von GnuPG mit Anwendungen

14.4.1 Benutzung von GnuPG mit Mutt

Fügen Sie folgendes zu `~/.muttrc` hinzu, um das langsame GnuPG vom automatischen Starten abzuhalten, während es beim Drücken von 'S' im Indexmenü gestartet werden kann.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

14.4.2 Benutzung von GnuPG mit Vim

Fügen Sie den Inhalt von `vimgpg` aus dem Beispielverzeichnis (`examples/`) in `~/.vimrc` ein, um GnuPG transparent zu starten.

Kapitel 15

Unterstützung für Debian

Die folgenden Ressourcen bieten Hilfe, Rat und Unterstützung für Debian an. Versuchen Sie Ihr bestes bei der Suche nach Hilfe bevor Sie sich an die Mailing-Listen wenden. :)

Auf eine Vielzahl von Dokumentationen in Ihrem System kann mittels eines WWW Browsers und den `dwww` oder `dhelp` Kommandos zugegriffen werden, die in den entsprechenden Paketen gefunden werden können.

15.1 Referenzen

Die folgenden Referenzen sind generell für Debian und Linux verfügbar. Wenn der Inhalt sich widerspricht, vertrauen Sie mehr den primären Informationsquellen als sekundären wie diesem Dokument.

- Installationshandbuch (primär)
  - Lesen Sie es bevor Sie installieren und upgraden.
  - Web: `http://www.debian.org/releases/stable/installmanual`
  - Web: `http://www.debian.org/releases/testing/installmanual` (befindet sich in Arbeit und ist manchmal eventuell nicht vorhanden)
  - Paket: `Not available in install-doc: Bug#155374`
  - Datei: `DebianCDunder/doc/`
- Release Notes (primär)
  - Dies müssen Sie vor der Installation und einem Upgrade unbedingt lesen, selbst wenn Sie erfahren sind.
  - Web: `http://www.debian.org/releases/stable/releasenotes`
  - Web: `http://www.debian.org/releases/testing/releasenotes` (befindet sich in Arbeit und ist manchmal eventuell nicht vorhanden)
  - Paket: `Not available in install-doc: Bug#155374`
  - Datei: `DebianCDunder/doc/`
- FAQ (sekundär)

  - Oft gestellte Fragen (FAQ, Frequently asked questions)
  - Web: http://www.debian.org/doc/manuals/debian-faq/
  - Paket: doc-debian
  - Datei: /usr/share/doc/debian/FAQ/index.html

- Debian Referenz (sekundär)
  - Das umfassendste Benutzerhandbuch für die Zeit nach der Installation
  - Web: http://www.debian.org/doc/manuals/debian-reference/
  - Paket: debian-reference
  - Datei: /usr/share/doc/Debian/reference/

- APT HOWTO (sekundär)
  - Detailliertes Benutzerhandbuch für das Debian Paketmanagement. (Woody)
  - Web: http://www.debian.org/doc/manuals/apt-howto/
  - Paket: apt-howto
  - Datei: /usr/share/doc/Debian/apt-howto/

- Debian Sicherheitshandbuch (sekundär)
  - Detailliertes Benutzerhandbuch zur Sicherung und Härtung der Standard Debian Installation. (Woody)
  - Web: http://www.debian.org/doc/manuals/securing-debian-howto/
  - Paket: harden-doc
  - Datei: /usr/share/doc/harden-doc/html/securing-debian-howto/

- dselect Dokumentation für Anfänger (sekundär)
  - Tutorial für dselect
  - Web: http://www.debian.org/releases/woody/i386/dselect-beginner
  - Paket: Not available in install-doc: Bug#155374
  - Datei: DebianCDunder/doc/

- Debian Policy Manual (primär)
  - Das technische Rückgrat von Debian.
  - Web: http://www.debian.org/doc/debian-policy/
  - Paket: debian-policy
  - Datei: /usr/share/doc/debian-policy/

- Debian Referenz für Entwickler (primär)
  - Grundwissen für Entwickler.
  - Die anderen sollten dies auch durchblättern.
  - Web: http://www.debian.org/doc/manuals/developers-reference/
  - Paket: developers-reference
  - Datei: /usr/share/doc/developers-reference/

- Debian Handbuch für neue Betreuer (primär)
  - Praktische Anleitung für Entwickler.
  - Tutorials zur Paketierung für den Rest von uns.

  - Web: `http://www.debian.org/doc/manuals/maint-guide/`
  - Paket: `maint-guide`
  - Datei: `/usr/share/doc/maint-guide/`

- Paketierungshandbuch (Potato)
  - `packaging-manual` Paket in Potato. (Wurde in den Anhang der *Referenz für Entwickler* verschoben.)

- Unix Manual-Pages (primär)
  - `man` *`Paket`*

- GNU info Seiten (primär)
  - `info` *`Paket`*

- Paket spezifische Dokumente (primär)
  - Diese sind unter `/usr/share/doc/`*`Paket`* zu finden

- LDP: Linux Dokumentations Projekt (sekundär)
  - Allgemeine Linux HOWTOs und mini-HOWTOs
  - Web: `http://www.tldp.org/`
  - Paket: `doc-linux-text`
  - Datei: `/usr/share/doc/HOWTO/`

- DDP: Debian Dokumentations Projekt (sekundär)
  - Debian spezifische Handbücher
  - Web: `http://www.debian.org/doc/`

- Debian Entwicklerecke (sekundär)
  - Schlüsselinformationen für Debian Entwickler
  - Einsichtig für Endbenutzer
  - Web: `http://www.debian.org/devel/`

- Quellcode (absolut primär)
  - Niemand kann dies bestreiten :-)
  - Laden Sie den Quellcode entsprechend ‘Der Quellcode’ auf Seite 11 herunter

Die folgenden Referenzen sind für Unix im allgemeinen verfügbar. Beachten Sie bitte, dass es einige kleinere Unterschiede zwischen verschiedenen Unix Systemen gibt. Gerätenamen und Initialisierungsmethoden bedürfen spezieller Beachtung.

- *The UNIX Programming Environment*
  - Das Buch das erklärt wie UNIX arbeitet.
  - Von B. W. Kernighan und R. Pike
  - Veröffentlicht von Princeton Hall Software Series

- *The C Programming Language* (zweite Ausgabe)
  - Das Buch über ANSI C.

  - Von B. W. Kernighan und D. M. Ritchie
  - Veröffentlicht von Princeton Hall Software Series

- *UNIX Power Tools*
  - Das Buch zu Unix Tipps.
  - Von Jerry Peek, Tim O'Reilly und Mike Loukides
  - Veröffentlicht von O'Reilly and Associates

- *Essential System Administration* (zweite Ausgabe)
  - Das Buch über Unix Systemadministration für viele Unixarten.
  - Von Aeleen Frisch
  - Veröffentlicht von O'Reilly and Associates

- Bell Labs: Computing Sciences Research
  - Reichhaltiges Archiv der Unix Geschichte
  - Main: `http://cm.bell-labs.com/cm/cs/`
  - Ausgewählte technische Berichte: `http://cm.bell-labs.com/cm/cs/cstr.html`
  - Einige Schriftstücke: `http://cm.bell-labs.com/cm/cs/papers.html`

- Allgemeine online Ressourcen zur Linux Unterstützung
  - Debian Planet (`http://www.debianplanet.org/`)
  - debianHELP (`http://www.debianhelp.org/`)
  - Linux.com (`http://linux.com/`)
  - Die Linux Homepage auf Linux Online (`http://www.linux.org/`)
  - Red Hat (kommerzieller Linux Verkäufer) (`http://www.redhat.com/`) (RPM, Sys-V Initialisierung)
  - SuSE, Inc. (kommerzieller Linux Verkäufer) (`http://www.suse.de/`) (RPM, Sys-V Initialisierung)
  - Slackware (`http://www.slackware.com/`) (TGZ, BSD artige Initialisierung)

- Allgemeine online Ressourcen zu Unix Anleitungen
  - Ein UNIX Einführungskurs der Ohio State University (`http://www-wks.acs.ohio-state.edu/unix_course/unix.html`)
  - UNIXhelp der University of Edinburgh (`http://unixhelp.ed.ac.uk/`)
  - Unix / Programmierungsinformationen (`http://arioch.unomaha.edu/~jclark/#info`)
  - comp.unix.questions FAQ (`http://www.faqs.org/faqs/unix-faq/faq/`)
  - comp.unix.user-friendly FAQ (`http://www.camelcity.com/~noel/usenet/cuuf-FAQ.htm`)
  - FreeBSD Dokumentation (`http://www.freebsd.org/docs.html`)
  - Das FreeBSD Handbuch (`http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html`)
  - UNIX ANLEITUNG (`http://ieee.uow.edu.au/documents/`)
  - Verein des Unix-Erbes (`http://www.tuhs.org/`)

- Homepages freier Software Projekte

- GNU Projekt (http://www.gnu.org/)
- Das Linux Dokumentations Projekt (http://www.tldp.org/)
- Die Linux Kernel Archive (http://www.linux.org/)
- Das XFree86 Projekt, Inc (http://www.xfree86.org/)
- GNOME (http://www.gnome.org/)
- K Desktop Environment (http://www.kde.org/)
- GNU Software bei Red Hat (http://sources.redhat.com/)
- Mozilla (http://www.mozilla.org)
- FreeBSD (http://www.freebsd.org/)
- OpenBSD (http://www.openbsd.org/)
- NetBSD (http://www.netbsd.org/)

15.2 Finden der Bedeutung eines Wortes

Viele in Debian verwendete Worte sind kryptisch oder Abkürzungen. Das folgende beantwortet die meisten Fragen:

```
$ dict sonderbares-Wort
```

15.3 Bestimmen der Popularität eines Debian Pakets

Viele Pakete existieren in Debian und es ist manchmal schwierig zu wissen, was man zuerst ausprobieren sollte. Vergleichen Sie die Ergebnisse des Debian Popularitäts Wettbewerbs (http://www.debian.org/~apenwarr/popcon/), für einen Überblick was andere nutzen. Installieren Sie das `popularity-contest` Paket, um dazu beizutragen.

15.4 Das Debian Fehlerverfolgungssystem

Die Debian Distribution hat ein Fehlerverfolgungssystem (http://bugs.debian.org/) (bug tracking system, BTS), das Details von Fehlerberichten enthält, die von Nutzern und Entwicklern gemeldet wurden. Jedem Fehler ist eine Nummer zugeordnet, und wird solange aufbewahrt, bis er als erledigt markiert wurde.

Sie sollten überprüfen, ob Ihr Fehlerbericht bereits von jemand anderem eingereicht wurde, bevor Sie ihn übertragen. Listen von aktuell ausstehenden Fehlern sind im World Wide Web (http://bugs.debian.org/) und sonstwo (http://www.debian.org/Bugs/Access) verfügbar. Vergleichen Sie auch 'Überprüfen von Bugs in Debian und Suche nach Hilfe' auf Seite 72.

Es sind viele Berichte zu ausgabekritischen Fehlern mit **FTBFS** markiert. Dies bedeutet "kann nicht aus dem Quellcode erzeugt werden" ("Fails To Build From Source").

Die Methode zum Berichten von Fehlern wird unter http://www.debian.org/Bugs/Reporting beschrieben.

15.5 Mailing-Listen

Lesen Sie zumindest "debian-devel-announce" (englisch, nur lesbar und geringes Aufkommen), um mit Debian auf dem Laufenden zu bleiben.

Die für Debian Nutzer am interessantesten Mailing-Listen sind "debian-user" (englisch, offen und hohes Aufkommen) und andere "debian-user-*Sprache*" Listen (für andere Sprachen).

Für Informationen zu diesen Listen und Details zur Abonnierung vergleichen Sie `http://lists.debian.org/`. Bitte überprüfen Sie die Archive für Antworten zu Ihren Fragen bevor Sie Anfragen stellen und achten Sie auf für die Liste geeignete Umgangsformen.

15.6 Internet Relay Chat (IRC)

IRC (Internet Relay Chat) ist ein Weg um sich mit Personen in der gesamten Welt in Echtzeit zu unterhalten. IRC Kanäle die sich Debian widmen können im freenode (`http://www.freenode.info/`) IRC Netzwerk gefunden werden. Zur Verbindung benötigen Sie einen IRC Client. Einige der populärsten Clients sind XChat, BitchX, ircII, irssi, epic4 und KSirc, all diese sind für Debian paketiert. Haben Sie den Client installiert, müssen Sie ihn auffordern, sich mit dem Server zu verbinden. In vielen Clients können Sie das durch Eingabe von

```
/server irc.debian.org
```

erreichen. Sind Sie verbunden, öffnen Sie den #debian Kanal mit

```
/join #debian
```

Bemerkung: Clients wie XChat haben oft eine unterschiedliche graphische Benutzerschnittstelle zum Verbinden mit Servern/Kanälen.

15.7 Suchmaschinen

Es gibt viele Suchmaschinen, die Dokumentationen zu Debian anbieten:

- Debian WWW Suchseite (`http://search.debian.org/`).
- Google (`http://www.google.com/`): fügen Sie "site:debian.org" als einen Suchbegriff ein.
- Google Groups (`http://groups.google.com/`): eine Suchmaschine für Newsgroups. Fügen Sie "group:linux.debian.*" als Suchbegriff ein.
- AltaVista (`http://www.altavista.com/`)

Eine Suche nach “cgi-perl” ergibt umfangreichere Erklärungen zu diesem Paket als die kurze Beschreibung in der Kontrolldatei. Vergleichen Sie ‘Überprüfen von Bugs in Debian und Suche nach Hilfe’ auf Seite 72 für ähnliche Ratschläge.

15.8 Webseiten

Es folgen einige URLs, die ich für spezielle Probleme sammelte.

- IBM developerWorks: Linux (http://www.ibm.com/developerworks/linux/)
- Adrian Bunk’s aktuellste Pakete (Backports nach stable) (http://www.fs.tum.de/~bunk/)
- Linux auf Laptops (http://www.linux-laptop.net/)
- Xterm FAQ (http://dickey.his.com/xterm/xterm.faq.html)
- EXT3 Dateisystem mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html)
- Unterstützung für große Dateien in Linux (http://www.suse.de/~aj/linux_lfs.html)
- Window Manager für X (http://www.xwinman.org)
- Linux USB Projekt (http://www.linux-usb.org/)
- SuSE Seiten für CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html)
- LNX-BBC (Das visitenkartengroße Boot CD Projekt) (http://www.lnx-bbc.org/)
- Linux Informationen von Karsten Self (Partitionierung, Backup, Browser, ...) (http://kmself.home.netcom.com/Linux/)
- Backup Informationen HOWTO von Alvin Oga (http://www.Linux-Backup.net/)
- Sicherheitsinformationen HOWTO von Alvin Oga (http://www.Linux-Sec.net/)
- Verchiedene INOFFIZIELLE Quellen für APT (http://www.apt-get.org/)
- Laptop Ethernet Konfiguration (http://www.orthogony.com/gjw/lap/lap-ether-intro.html)

Anhang A

Anhang

A.1 Autoren

Die Debian Referenz wurde ursprünglich von Osamu Aoki <osamu@debian.org> als eine persönliche Installationsanleitung geschrieben, die "Schnellreferenz ..." hieß. Ein großer Teil des Inhalts kam vom Archiv der "debian-user" Mailing-Liste. Auch "Debian Installation Manual" und "Debian Release Notes" waren Referenzen.

Einem Vorschlag von Josip Rodin folgend, der sehr aktiv im Debian Dokumentations Projekt (http://www.debian.org/doc/ddp) (DDP) ist und der der aktuelle Betreuer der "Debian FAQ" ist, wurde dieses Dokument in "Debian Referenz" umbenannt und einige Kapitel der "Debian FAQ" mit verweisartigem Inhalt hinzugefügt. Danach wurde "Debian Schnellreferenz" als ein Auszug geschaffen.

Dieses Dokument wurde geschrieben, übersetzt und erweitert durch die folgenden QREF Teammitglieder:

- Englisches Original für die ursprüngliche "Schnellreferenz ..."
  - Osamu Aoki <osamu@debian.org> (Koordinator des gesamten Inhalts)
- Englisches Korrekturlesen und zusätzliche Beiträge
  - David Sewell <dsewell@virginia.edu> (Koordinator en Stil)
  - Thomas Hood <jdthood@yahoo.co.uk> (Netzwerk bezogen)
  - Brian Nelson <nelson@bignachos.com> (besonders X-bezogen)
  - Jan Michael C Alonzo <jmalonzo@spaceants.net>
  - Daniel Webb <webb@robust.colorado.edu>
  - Rückmeldungen von allen Übersetzern
- Französische Übersetzung
  - Guillaume Erbs <gerbs@free.fr> (Koordinator fr)
  - Rénald Casagraude <rcasagraude@interfaces.fr>
  - Jean-Pierre Delange <delange@imaginet.fr>
  - Daniel Desages <daniel@desages.com>
- Italienische Übersetzung
  - Davide Di Lazzaro <mc0315@mclink.it> (Koordinator it)
- Portugiesische Übersetzung (brasilianisches Portugiesisch)

  - Paulo Rogério Ormenese <pormenese@uol.com.br> (Koordinator pt-br)
  - Andre Luis Lopes <andrelop@ig.com.br>
  - Marcio Roberto Teixeira <marciotex@pop.com.br>
  - Rildo Taveira de Oliveira <to_rei@yahoo.com>
  - Raphael Bittencourt Simoes Costa <raphael-bsc@bol.com.br>
  - Gustavo Noronha Silva <kov@debian.org> (Koordinator)
- Spanische Übersetzung
  - Walter Echarri <wecharri@infovia.com.ar> (Koordinator es)
  - José Carreiro <ffx@urbanet.ch>
- Deutsche Übersetzung
  - Jens Seidel <tux-master@web.de> (Koordinator de)
  - Willi Dyck <wdyck@gmx.net>
  - Stefan Schröder <stefan@fkp.uni-hannover.de>
  - Agon S. Buchholz <asb@kefk.net>
- Polnische Übersetzung – Mitglieder von PDDP (http://debian.linux.org.pl):
  - Marcin Andruszkiewicz
  - Mariusz Centka <mariusz.centka@debian.linux.org.pl>
  - Bartosz Fenski <fenio@debian.linux.org.pl> (Koordinator pl)
  - Radoslaw Grzanka <radekg@debian.linux.org.pl>
  - Bartosz 'Xebord' Janowski
  - Jacek Lachowicz
  - Rafal Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierala <zen@debian.linux.org.pl>
  - Oskar Ostafin <cx@debian.linux.org.pl>
  - Tomasz Piekos
  - Jacek Politowski
  - Mateusz Prichacz <mateusz@debian.linux.org.pl>
  - Marcin Rogowski
  - Pawel Rozanski
  - Mariusz Strzelecki
  - Krzysztof Scierski
  - Przemyslaw Adam Smiejek <tristan@debian.linux.org.pl>
  - Mateusz Tryka <uszek@debian.linux.org.pl>
  - Cezary Uchto
  - Krzysztof Witkowski <tjup@debian.linux.org.pl>
  - Bartosz Zapalowski <zapal@debian.linux.org.pl>
- Chinesische Übersetzung (vereinfachtes Chinesisch)
  - Hao "Lyoo" LIU <iamlyoo@163.net> (Koordinator zh-cn)
  - Ming Hua <minghua@rice.edu>
- Chinesische Übersetzung (traditionelles Chinesisch)
  - Tang Wei Ching <wctang@csie.nctu.edu.tw> (Koordinator zh-tw)

QREF ist die Kurzform für den ursprünglichen Dokumenttitel "Quick Reference…" ("Schnellreferenz …") und ist auch der Projektname auf qref.sourceforge.net.

Der Großteil des Inhalts von 'Debian Grundlagen' auf Seite 5 kam ursprünglich von der

“Debian FAQ” (März 2002):

- 5. Die Debian FTP Archive: `ftparchives.sgml` (gesamtes Kapitel)
- 6. Grundlagen des Debian Paketmanagement Systems: `pkg_basics.sgml` (gesamtes Kapitel)
- 7. Die Debian Paketmanagement Tools: `pkgtools.sgml` (gesamtes Kapitel)
- 8. Aktuell halten des Debian Systems: `uptodate.sgml` (gesamtes Kapitel)
- 9. Debian und der Kernel: `kernel.sgml` (gesamtes Kapitel)
- 10. Anpassen Ihrer Installation von Debian GNU/Linux: `customizing.sgml` (Teil des Kapitels)

Diese Abschnitte der “Debian FAQ” wurden in diesem Dokument nach großer Umorganisation eingebunden, um Änderungen im Debian System zu folgen. Beide Dokumente werden nun gleichzeitig aktualisiert.

Die ursprüngliche “Debian FAQ” wurde erstellt und betreut von J. H. M. Dassen (Ray) und Chuck Stickelman. Autoren der umgeschriebenen “Debian FAQ” sind Susan G. Kleinmann und Sven Rudolph. Nach diesen wurde die “Debian FAQ” von Santiago Vila betreut. Der aktuelle Betreuer ist Josip Rodin.

Teile der Informationen für die “Debian FAQ” kommen von:

- Der Debian-1.1 Ausgabe Ankündigung von Bruce Perens (`http://www.perens.com/`).
- Der Linux FAQ von Ian Jackson (`http://www.chiark.greenend.org.uk/~ijackson/`).
- Dem Debian Mailing-Listen Archiv (`http://lists.debian.org/`).
- Dem dpkg Programmierer Handbuch und das Debian Policy Handbuch (vergleichen Sie ‘Referenzen’ auf Seite 199)
- vielen Entwicklern, Freiwilligen und Beta Testern sowie
- dem unzureichenden Gedächtnis der Autoren. :-)

Die Autoren möchten all jenen danken, die halfen dieses Dokument zu ermöglichen.

A.2 Haftung

Da ich kein Experte bin, gebe ich nicht vor, ein volles Wissen über Debian oder Linux im allgemeinen zu besitzen. Die Sicherheitsüberlegungen die ich aufführe sind eventuell nur für private Anwendungen anwendbar.

Dieses Dokument ersetzt nicht die ultimativen Anleitungen.

Jegliche Haftung wird abgelehnt. Alle Handelsmarken sind das Eigentum der entsprechenden Markeninhaber.

A.3 Rückmeldungen

Kommentare und Ergänzungen zu diesem Dokument sind immer willkommen. Senden Sie bitte eine E-Mail an das Debian BTS System (`http://bugs.debian.org/`) unter dem

`debian-reference` Paket oder unter dem entsprechend übersetzten Paket. Das Verwenden von `reportbug` vereinfacht das Erstellen eines vollständigen Fehlerberichts. Sie können auch in englisch eine E-Mail an Osamu Aoki (`http://people.debian.org/~osamu/`) nach `<osamu@debian.org>` oder zu jedem Übersetzer in dessen Sprache senden.

Obwohl ich in den USA lebte, bin ich kein einheimischer englischer Nutzer. Jegliche grammatikalische Korrekturen sind willkommen.

Die beste Rückmeldung ist ein Patch gegen die SGML Version, aber ein Patch gegen die Textversion ist auch willkommen. Vergleichen Sie 'Das offizielle Dokument' auf Seite 1 für die offizielle Dokumentseite.

Die ursprünglichen SGML Dateien, die zur Erzeugung dieses Dokuments verwendet wurden, sind auch im CVS erhältlich unter: `:pserver:anonymous@cvs.qref.sf.net/cvsroot/qref` oder `http://qref.sourceforge.net/Debian/qref.tar.gz`.

A.4 Dokumentformat

Dieses Dokument wurde mit der DebianDoc SGML DTD geschrieben (umgeschrieben von LinuxDoc SGML). Das DebianDoc SGML System ermöglicht es, Dateien in verschiedenen Formaten aus einer Quelle zu erzeugen, z.B. kann dieses Dokument in HTML, einfachem Text, TeX DVI, PostScript, PDF und GNU info betrachtet werden.

Konvertierungshilfsmittel für DebianDoc SGML sind im Debian Paket `debiandoc-sgml` erhältlich.

A.5 Das Debian Labyrinth

Das Linux System ist eine sehr mächtige Computerplattform für einen vernetzten Computer. Dennoch ist das Lernen wie man all diese Eigenschaften nutzt nicht einfach. Das Einrichten eines Druckers ist ein gutes Beispiel.

Es gibt ein komplettes, detailliertes Verzeichnis namens "QUELLCODE". Dies ist sehr akkurat aber sehr schlecht zu verstehen. Es gibt auch Referenzen namens HOWTO und mini-HOWTO. Diese sind einfacher zu verstehen, tendieren aber dazu, zu viele Details anzugeben, so dass man das große Bild aus den Augen verliert. Ich habe manchmal das Problem, einen bestimmten Abschnitt in einem langen HOWTO zu finden, wenn ich nach einigen Kommandos suche.

Um durch dieses Labyrinth der Linux Systemkonfiguration zu navigieren, begann ich einige Notizen zur Erinnerung im Textformat zu meiner schnellen Referenz zu schreiben. Diese Liste von Notizen wuchs schnell und ich lernte debiandoc in der Zwischenzeit. Das Ergebnis ist diese *Debian Referenz*.

A.6 Debian Zitate

Es folgen einige interessante Zitate von der Debian Mailing-Liste.

- "Dies ist Unix. Es gibt dir genug Seil, um dich selbst aufzuhängen." – Miquel van Smoorenburg <miquels@cistron.nl>
- "Unix **IST** benutzerfreundlich. . . Es ist einfach nur wählerisch, wer seine Freunde sind." – Tollef Fog Heen <tollef@add.no>